

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

«До захисту допущено»

В. о. завідувача кафедри

О.Л.Тимошук

«___» _____ 20 __ р.

Дипломна робота

на здобуття ступеня бакалавра з напрямку підготовки 6.040303 «Системний
аналіз» на тему: «Системи стабілізації нестійких режимів в соціально-
навчальному процесі студента»

Виконав студент IV курсу, групи КА-54

Богдан Давид Романович

Керівник:

професор кафедри ММСА, д.т.н., проф.

Романенко В.Д.

Консультант з економічного розділу:

доцент кафедри теоретичної та прикладної

економіки КПІ ім. Ігоря Сікорського, к.е.н., доц.

Шевчук О.А.

Консультант з нормоконтролю:

доцент кафедри ММСА, к.т.н., доц.

Коваленко А.Є.

Рецензент:

доцент кафедри системного аналізу та теорії

прийняття рішень Київського національного

університету імені Тараса Шевченка, к.ф.-м.н., доц.

Зінько П.М.

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань

Студент

Київ
2019

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — перший (бакалаврський)
Напрямок підготовки — 6.040303 «Системний аналіз»

ЗАТВЕРДЖУЮ

В. о. завідувача кафедри

О.Л.Тимошук

«___» _____ 20 __ р.

ЗАВДАННЯ

на дипломну роботу студенту Богдану Давиду Романовичу

1. Тема роботи «Системи стабілізації нестійких режимів в соціально-навчальному процесі студента», керівник роботи Романенко Віктор Демидович, д.т.н., проф., затверджені наказом по університету від 25 травня 2019 року № 1353-с

2. Термін подання студентом роботи _____

3. Вихідні дані до роботи _____

4. Зміст роботи _____

5. Перелік ілюстративного матеріалу _____

6. Консультант з економічного розділу роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ 4. Функціонально-вартісний аналіз програмного продукту	Шевчук О.А., доцент кафедри теоретичної та прикладної економіки КПІ ім. Ігоря Сікорського		

7. Дата видачі завдання _____

Календарний план

№/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи

Студент

Д.Р. Богдан

Керівник роботи

В.Д. Романенко

РЕФЕРАТ

Дипломна робота: 90 с., 22 рис., 6 табл., 2 додатки, 12 джерел.

КОГНІТИВНА КАРТА, СОЦІАЛЬНО-НАВЧАЛЬНИЙ ПРОЦЕС СТУДЕНТА, НЕСТІЙКИЙ РЕЖИМ, КЕРУВАННЯ, СТАБІЛІЗАЦІЯ.

Дана дипломна робота присвячена дослідженню нестійкого режиму соціально-навчального процесу студента і методів його стабілізації.

За допомогою когнітивної карти, що представляє собою зважений орієнтований граф, було представлено соціально-навчальний процес студента, який має нестійкий характер. Для стабілізації нестійкого імпульсного процесу когнітивної карти, було здійснено перехід до моделі у просторі станів і застосовано методи теорії керування, що базуються на еталонних моделях. Також, на випадок невідомих значень вагів когнітивної карти, для оцінки параметрів моделі було застосовано рекурентний метод найменших квадратів.

ABSTRACT

The work consist of 90 p., 22 fig., 6 tables, 2 append, 12 sources.

The theme: «Systems of stabilization of unstable regimes in the social and educational process of a student».

COGNITIVE MAP, SOCIAL-EDUCATIONAL PROCESS OF STUDENT, UNSTABLE REGIMES, CONTROL SYSTEM, STABILIZATION.

This work is devoted to the study of the unstable regime of the student's social and educational process and the methods of its stabilization.

With the help of a cognitive map representing a weighted tentative graph, the student's socio-educational process was presented, which is unstable. To stabilize the unstable impulse process of a cognitive map, a transition to a model in the state space was made and methods of control theory based on reference models were applied. Also, in the case of unknown values of the weights of a cognitive map, a recursive least squares method was used to evaluate the model parameters.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	8
ВСТУП.....	9
РОЗДІЛ 1 ЗАБЕЗПЕЧЕННЯ СТІЙКОСТІ ІМПУЛЬСНИХ ПРОЦЕСІВ В КОГНІТИВНИХ КАРТАХ НА ОСНОВІ МОДЕЛЕЙ У ПРОСТОРИ СТАНІВ	11
1.1 Визначення та позначення	11
1.2 Перехід від моделі розімкнутої системи в просторі станів до моделі на основі когнітивної карти	12
1.3 Перехід від моделі замкненої системи з регулятором стану до моделі, представленої у вигляді когнітивної карти	18
1.4 Керування в когнітивній моделі зі спостерігачем стану	22
1.5 Перехід від когнітивної моделі до моделі в просторі станів.....	24
1.6 Висновки до розділу 1	27
РОЗДІЛ 2 СИСТЕМИ СТАБІЛІЗАЦІЇ. КОГНІТИВНА КАРТА СОЦІАЛЬНО-НАВЧАЛЬНОГО ПРОЦЕСУ СТУДЕНТА	28
2.1 Когнітивна карта соціально-навчального процесу студента	28
2.2 Розробка системи стабілізації нестійкого імпульсного процесу когнітивної карти в приростах змінних.....	31
2.3 Розробка системи адаптивного керування імпульсним процесом при невідомих і змінних коефіцієнтах когнітивної карти	36
2.4 Розробка системи стабілізації координат вершин когнітивної карти на заданих рівнях при нестійкому імпульсному процесі	38
2.5 Висновки до розділу 2	40
РОЗДІЛ 3 ЧИСЕЛЬНЕ МОДЕЛЮВАННЯ. АНАЛІЗ РЕЗУЛЬТАТІВ	41
3.1 Моделювання нестійкого імпульсного режиму соціально-навчального процесу студента.....	41

3.2 Система стабілізації соціально-навчального процесу студента	43
3.3 Система стабілізації соціально-навчального процесу студента при невідомій матриці суміжності	46
3.4 Система стабілізації соціально-навчального процесу студента на заданих рівнях.....	51
3.5 Висновки до розділу 3	53
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ	55
4.1 Постановка задачі проектування	56
4.1.1 Обґрунтування функцій програмного продукту	56
4.1.2 Варіанти реалізації основних функцій	57
4.2 Обґрунтування системи параметрів програмного продукту	60
4.2.1 Опис параметрів.....	60
4.2.2 Кількісна оцінка параметрів	61
4.2.3 Аналіз експертного оцінювання параметрів	64
4.3 Аналіз рівня якості варіантів реалізації функцій	68
4.4 Економічний аналіз варіантів розробки програмного продукту	70
4.5 Вибір кращого варіанта програмного продукту техніко-економічного рівня	76
4.6 Висновки до розділу 4	76
ВИСНОВКИ	78
ЛІТЕРАТУРА.....	79
ДОДАТОК А ІЛЮСТРАТИВНИЙ МАТЕРІАЛ ДОПОВІДІ	81
ДОДАТОК Б КОД ПРОГРАМНОГО ПРОДУКТУ	89

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

КК – когнітивна карта

ПП – програмний продукт

ФВА – функціонально-вартісний аналіз

ВСТУП

Соціально-навчальний процес студента — це складна динамічна система, яка включає в себе багато факторів, що взаємодіють між собою та змінюються в часі. Це пов'язано з тим, що, крім навчання, студент, як правило, має родину, роботу (особливо на старших курсах), хобі тощо. У той же час часові, фізичні, матеріальні та інтелектуальні ресурси студента обмежені, і тому, наприклад, збільшення часу, що виділяється студентом на роботу, призводить до зменшення успіхів у навчанні. Таким чином, навіть у першому наближенні, для опису життя студента потрібно застосувати модель, яка б враховувала численні взаємозв'язки між окремими факторами як у статиці, так і в динаміці. Когнітивна карта найкраще підходить на цю роль.

Згідно з [1, 2], когнітивна карта – це орієнтований граф, вершини (вузли) якого відображають деякі фактори (поняття, сутності, концепти), а ребра – зв'язки між цими факторами. Аксельрод [1] використовував цю методологію переважно в психології (звідси й назва), а Робертс [2] переважно розвивав математичний апарат теорії графів для вирішення соціально-економічних задач. Першим запропонованим варіантом когнітивної карти є знаковий орграф, в якому знак «плюс» над ребром з вершини А у вершину Б вказує на те, що збільшення значення у вершині А призведе до збільшення значення у вершині Б, а знак «мінус» вказує, що збільшення в А веде до зменшення у Б. Проте, більш цікавим є випадок зваженого орграфа, який також запропонував Робертс [2], коли над ребрами надписуються не знаки, а числа, що характеризують ступінь впливу фактору А на фактор Б.

У наш час зарубіжні вчені приділяють багато уваги дослідженню нечітких когнітивних карт і багатьох інших модифікацій [5, 6]. Модифікації включають в себе функціональні когнітивні карти (над ребрами надписуються

функції), динамічні когнітивні карти, карти з запізненням та ін. Слід зазначити, що більшість досліджень присвячено побудові, моделюванню, аналізу, застосуванню когнітивних карт, але задача стабілізації і керування у когнітивному моделюванні залишається ще здебільше не вирішеною.

Перший розділ показує як когнітивна модель співвідноситься з моделлю у просторі станів, та навпаки. Наводяться методи переходу від однієї моделі до іншої.

У другому розділі будується когнітивна карта соціально-навчального процесу студента, та розробляються системи стабілізації нестійкого імпульсного процесу КК за допомогою регулятора стану, що базується на еталонних моделях. На випадок невідомих параметрів моделі було розглянуто рекурентний метод найменших квадратів для їх оцінки. Також розробляється система стабілізації вершин КК на заданих рівнях.

Третій розділ містить результати моделювання динаміки КК соціально-навчального процесу студента, стабілізації її імпульсного процесу за допомогою регулятора стану, що базується на еталонних моделях, оцінки рекурентним методом найменших квадратів параметрів моделі та стабілізації вершин КК на заданих рівнях за допомогою регулятора стану, що, також, базується на еталонних моделях.

Четвертий розділ присвячено функціонально-вартісному аналізу розробленого програмного продукту.

РОЗДІЛ 1

ЗАБЕЗПЕЧЕННЯ СТІЙКОСТІ ІМПУЛЬСНИХ ПРОЦЕСІВ В КОГНІТИВНИХ КАРТАХ НА ОСНОВІ МОДЕЛЕЙ У ПРОСТОРИ СТАНІВ

1.1 Визначення та позначення

Введемо деякі позначення, які будемо використовувати. Нехай когнітивна модель буде задана зваженим орієнтованим графом $\langle Z, W \rangle$, де Z – множина із n вершин (факторів) з числовими значеннями в кожній вершині; W – множина орієнтованих ребр з вагами. Будемо позначати також через $Z(k)$ вектор значень $z_i(k)$, $i = 1, \dots, n$, у всіх вершинах в момент часу k , через W – матрицю суміжності з вагами зв'язків, w_{ij} , $i = 1, \dots, n$, $j = 1, \dots, n$, – вага зв'язку ребра, що виходить із вершини x_i в вершину x_j . В [2] Робертс розглядає «правило зміни значень параметрів вершин», тобто математичний вираз ступеня впливу зміни одного фактору на інший. Найбільш поширене (але не єдине) правило зміни значень параметрів вершин формуються у виді різницевого рівняння першого порядку в приростах [11]

$$\Delta z_i(k+1) = \sum_{j=1}^n w_{ij} \Delta z_j(k). \quad (1.1)$$

де $\Delta z_i(k) = z_i(k) - z_i(k-1)$, $i = 1, \dots, n$.

В векторній формі (1.1) можна записати як

$$\Delta Z(k+1) = W^T \Delta Z(k). \quad (1.2)$$

В подальшому ми будемо використовувати саме це правило. Відмитим, що в правій частині рівності (1.1) може знаходитись доданок з $\Delta z_i(k)$, тобто в графі

можливі петлі (приріст в даній вершині може залежати від приросту в цій же вершині на минулому кроці).

Одним з важливіших понять при аналізі когнітивних карт є поняття стійкості. Існує багато визначень і критеріїв стійкості когнітивних моделей. Ми будемо використовувати наступні найбільш загальні і прості визначення [2]. Вершина Z_i називається імпульсно стійкою, якщо послідовність $\{|\Delta Z_i(k)|: k = 0, 1, 2, \dots\}$ обмежена при імпульсному процесі (1.1), визначеним довільним набором початкових значень і імпульсів (приростів). Вершина Z_i називається абсолютно стійкою, якщо послідовність $\{|Z_i(k)|: k = 0, 1, 2, \dots\}$ обмежена при довільному імпульсному процесі (1.1). Когнітивна карта називається імпульсно (абсолютно) стійкою, якщо всі її вершини імпульсно (абсолютно) стійкі.

1.2 Перехід від моделі розімкнутої системи в просторі станів до моделі на основі когнітивної карти

Розглянемо для початку простішу двовимірну модель динаміки процесу в просторі станів

$$\begin{aligned}x_1(k+1) &= f_{11}x_1(k) + f_{12}x_2(k) + g_1u(k), \\x_2(k+1) &= f_{21}x_1(k) + f_{22}x_2(k) + g_2u(k), \\y(k) &= c_1x_1(k) + c_2x_2(k).\end{aligned}\tag{1.3}$$

Покажемо яким чином її можна зобразити у вигляді когнітивної карти, описаною моделлю (1.1). По-перше, відмітим, що модель не зміниться, якщо

записати її в приростах (звичайно, при умові відповідного вибору початкових значень)

$$\begin{aligned}\Delta x_1(k+1) &= f_{11}\Delta x_1(k) + f_{12}\Delta x_2(k) + g_1\Delta u(k), \\ \Delta x_2(k+1) &= f_{21}\Delta x_1(k) + f_{22}\Delta x_2(k) + g_2\Delta u(k), \\ \Delta y(k) &= c_1\Delta x_1(k) + c_2\Delta x_2(k).\end{aligned}\tag{1.4}$$

Основна відміна моделі (1.4) від моделі (1.1) полягає в тому, що в (1.1) неможливо відобразити миттєві зв'язки – рівняння виміру (1.4). Перетворимо (1.4) наступним чином: напишемо рівняння виміру на один крок вперед, а потім підставимо в нього рівняння стану. Отримаємо

$$\begin{aligned}\Delta x_1(k+1) &= f_{11}\Delta x_1(k) + f_{12}\Delta x_2(k) + g_1\Delta u(k), \\ \Delta x_2(k+1) &= f_{21}\Delta x_1(k) + f_{22}\Delta x_2(k) + g_2\Delta u(k), \\ \Delta y(k+1) &= (c_1f_{11} + c_2f_{21})\Delta x_1(k) + (c_1f_{12} + c_2f_{22})\Delta x_2(k) + \\ &\quad + (c_1g_1 + c_2g_2)\Delta u(k).\end{aligned}\tag{1.5}$$

Нехай $Z = (u \ x_1 \ x_2 \ y)^T$. Тут і далі будемо припускати саме таку нумерацію вершин в графі: спочатку вершини керування, потім стани і спостерігання. Представимо (1.5) у формі (1.2). В результаті отримаємо когнітивну карту, зображену на рис. 1.1 з такою ваговою матрицею суміжності [11]

$$W = \begin{pmatrix} 0 & g_1 & g_2 & c_1g_1 + c_2g_2 \\ 0 & f_{11} & f_{21} & c_1f_{11} + c_2f_{21} \\ 0 & f_{12} & f_{22} & c_1f_{12} + c_2f_{22} \\ 0 & 0 & 0 & 0 \end{pmatrix}.\tag{1.6}$$

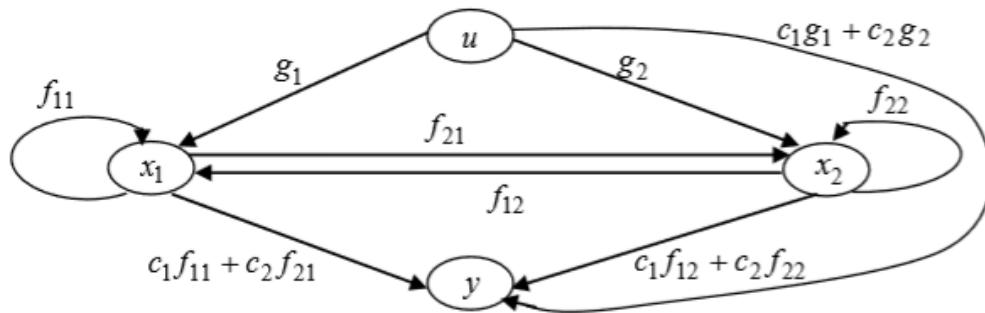


Рисунок 1.1

Абсолютна стійкість графа, заданого матрицею (1.6), досягається, коли всі її власні числа по модулю менше одиниці [2]. Розкриваючи $\det(zI_4 - W)$, де I_4 – одинична матриця четвертого порядку, легко побачити, що матриця W має два нульових власних числа, а інші два – це власні числа матриці $\begin{pmatrix} f_{11} & f_{21} \\ f_{12} & f_{22} \end{pmatrix}$. Отже, абсолютна стійкість когнітивної карти забезпечується асимптотичною стійкістю системи в просторі станів.

Покажемо, що довільну модель динаміки в просторі станів можна представити у вигляді когнітивної моделі. Нехай модель має вигляд

$$X(k+1) = FX(k) + GU(k), \quad Y(k) = CX(k). \quad (1.7)$$

Теорема 1. Динаміка процесу в просторі станів (1.7) може бути представлена у вигляді когнітивної карти з ваговою матрицею суміжності W , де

$$W^T = \begin{pmatrix} 0 & 0 & 0 \\ G & F & 0 \\ CG & CF & 0 \end{pmatrix}. \quad (1.8)$$

Доведення. Зобразимо (1.7) в приростах змінних

$$\Delta X(k+1) = F\Delta X(k) + G\Delta U(k),$$

$$\Delta Y(k) = C\Delta X(k),$$

$$\Delta X(k) = X(k) - X(k-1),$$

$$\Delta U(k) = U(k) - U(k-1),$$

$$\Delta Y(k) = Y(k) - Y(k-1).$$

Розглянемо рівняння виміру на один крок квантування вперед і підставимо в нього праву частину рівняння стану, після чого отримаємо:

$$\begin{aligned}\Delta X(k+1) &= F\Delta X(k) + G\Delta U(k), \\ \Delta Y(k+1) &= CF\Delta X(k) + CG\Delta U(k).\end{aligned}\tag{1.9}$$

Безпосереднє обчислення дозволяє переконатись, що для того, аби отриманий вираз (1.9) був еквівалентним (1.7), достатньо покласти

$$X(0) = X_0,$$

$$Y(0) = Y_0 = CX_0,$$

$$U(0) = U_0,$$

$$\Delta X(1) = (F - I)X_0 + GU_0,$$

$$\Delta U(1) = 0,$$

$$\Delta Y(1) = C\Delta X(1),$$

де X_0, U_0, Y_0 – початкові значення моделі (1.7).

Передбачається, що на систему в подальшому не діють зовнішні імпульси, у тому числі й керуючі, тобто $U(k) = U_0 \forall k \geq 0$, або інакше $\Delta U(k) = 0 \forall k \geq 0$.

Випадок, коли керування формується в замкнутій системі, буде розглянуто нижче.

Нехай в когнітивній карті кількість вершин рівна кількості змінних, в залежності від належності до векторів X, U, Y будемо називати їх вершинами стану, керування і спостерігання відповідно. Коефіцієнти матриці G являються вагами ребр ведучих від керуючих вершин до вершин станів. Коефіцієнти матриці F присвоюються ребрам, що поєднують вершини станів між собою. (діагональні елементи – ваги петель). Від вершин станів до вершин спостерігання ведуть ребра з вагами із коефіцієнтів матриці CF , а від вершин керування до вершин спостерігання – із коефіцієнтів матриці CG . Оскільки система розімкнена, й динаміка вектора керування не визначена системою (1.7), у вершин керування нема ребр що входять, тобто відповідні стовбці матриці суміжності нульові. Тоді

$$\begin{pmatrix} \Delta U(k+1) \\ \Delta X(k+1) \\ \Delta Y(k+1) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ G & F & 0 \\ CG & CF & 0 \end{pmatrix} \begin{pmatrix} \Delta U(k) \\ \Delta X(k) \\ \Delta Y(k) \end{pmatrix}.$$

Таким чином, виходячи із (1.2), матриця суміжності задовольняє (1.8). Теорему доведено.

Покажемо в загальному випадку відповідність між власними числами матриці стану і коренями характеристичного полінома когнітивної карти.

Теорема 2. Всі ненульові власні числа когнітивної карти, що описує (1.8), рівні власним числам матриці стану моделі (1.7).

Доведення. Розглянемо

$$\det(zI_n - W) = \det(zI_n - W^T) = \begin{vmatrix} zI_{n_u} & 0 & 0 \\ -G & zI_{n_x} - F & 0 \\ -CG & -CF & zI_{n_y} \end{vmatrix},$$

де n – розмірність одиничної матриці;

n_u, n_x, n_y – розмірності векторів керування, стану, спостереження відповідно ($n = n_u + n_x + n_y$).

Будемо розкривати визначник спочатку по строкам згори донизу і отримаємо

$$\det(zI_n - W) = z^{n_u} \begin{vmatrix} zI_{n_x} & 0 \\ -CF & zI_{n_y} \end{vmatrix}.$$

Потім будемо розкривати визначник по стовбцям справа наліво і отримаємо $\det(zI_n - W) = z^{n_u + n_y} \det(zI - F)$. Отже, корні характеристичного поліному матриці суміжності (її власні числа) містять нуль (кратності $n_u + n_y$) і корні поліному $\det(zI - F)$, тобто власні числа матриці стану F моделі (1.7). Теорему доведено.

Наслідок 1. Якщо власні числа матриці F по модулю менше одиниці (тобто модель (1.7) асимптотично стійка), то когнітивна модель, обумовлена матрицею суміжності (1.8), буде абсолютно стійкою.

Доведення. Когнітивна модель абсолютно стійка, якщо всі власні числа матриці суміжності по модулю менше одиниці. А із теорії керування відомо, що модель (1.7) асимптотично стійка тоді і тільки тоді, коли власні числа матриці стану по модулю менше одиниці. Звідси і із теореми 2 отримуємо необхідне твердження.

1.3 Перехід від моделі замкненої системи з регулятором стану до моделі, представленої у вигляді когнітивної карти

Найпростішим способом стабілізації динамічної моделі, заданої в просторі станів, являється регулятор стану. Розглянемо, яким чином систему з регулятором стану (при деякій заданій матриці посилення регулятора) можна представити у вигляді когнітивної карти [11].

Продовжимо минулий приклад, додав до (1.3) рівняння регулятора стану

$$\begin{aligned}x_1(k+1) &= f_{11}x_1(k) + f_{12}x_2(k) + g_1u(k), \\x_2(k+1) &= f_{21}x_1(k) + f_{22}x_2(k) + g_2u(k), \\y(k) &= c_1x_1(k) + c_2x_2(k), \\u(k) &= -K_1x_1(k) - K_2x_2(k).\end{aligned}\tag{1.10}$$

Після переходу к приростам необхідно «позбутися» від миттєвої взаємодії в регуляторі, не змінюючи при цьому динаміки системи. Розглянемо рівняння регулятора на один крок квантування вперед і підставимо в праву частину рівняння стану. Після перетворень будемо мати

$$\begin{aligned}\Delta x_1(k+1) &= f_{11}\Delta x_1(k) + f_{12}\Delta x_2(k) + g_1\Delta u(k), \\ \Delta x_2(k+1) &= f_{21}\Delta x_1(k) + f_{22}\Delta x_2(k) + g_2\Delta u(k), \\ \Delta y(k+1) &= (c_1f_{11} + c_2f_{21})\Delta x_1(k) + (c_1f_{12} + c_2f_{22})\Delta x_2(k) + \\ &\quad + (c_1g_1 + c_2g_2)\Delta u(k), \\ \Delta u(k+1) &= -(K_1f_{11} + K_2f_{21})\Delta x_1(k) - (K_1f_{12} + K_2f_{22})\Delta x_2(k) - \\ &\quad - (K_1g_1 + K_2g_2)\Delta u(k).\end{aligned}\tag{1.11}$$

Матриця суміжності має вид

$$W = \begin{pmatrix} -(K_1 g_1 + K_2 g_2) & g_1 & g_2 & c_1 g_1 + c_2 g_2 \\ -(K_1 f_{11} + K_2 f_{21}) & f_{11} & f_{21} & c_1 f_{11} + c_2 f_{21} \\ -(K_1 f_{12} + K_2 f_{22}) & f_{12} & f_{22} & c_1 f_{12} + c_2 f_{22} \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (1.12)$$

Отримаємо когнітивну карту з керуванням (рис. 1.2)

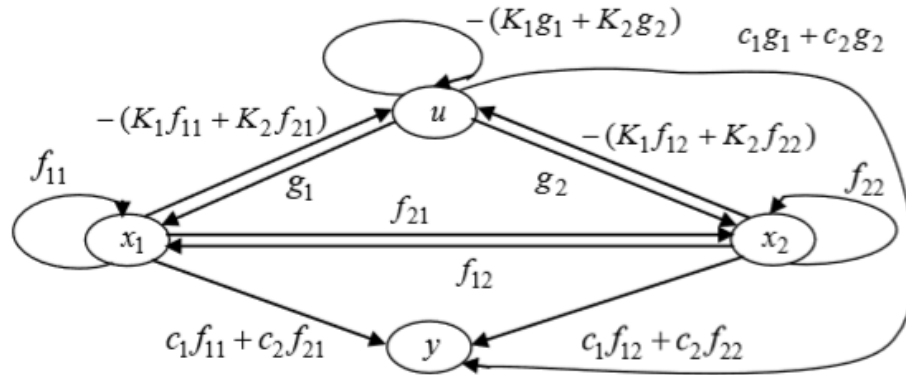


Рисунок 1.2 – Приклад когнітивної карти з керуванням

Знайдемо характеристичне рівняння цієї моделі (проміжні перетворення спростовані):

$$\det(zI_4 - W) = z^2(z^2 + z(K_1 g_1 + K_2 g_2 - f_{11} - f_{22}) + f_{11}f_{22} - f_{12}f_{21} + K_1 g_2 f_{12} + K_2 g_1 f_{21} - K_1 g_1 f_{22} - K_2 g_2 f_{11}). \quad (1.13)$$

Отже, рівняння (1.13) має два нульових кореня, а інші два кореня можна розмістити бажаним чином, обираючи відповідно K_1 , K_2 (метод синтезу регулятора по бажаному розміщенню полюсів). В даному випадку це легко зробити за допомогою теореми Вієта, а саме – нехай бажані полюси замкненої системи рівні z_1, z_2 , тоді

$$K_1 g_1 + K_2 g_2 - f_{11} - f_{22} = -(z_1 + z_2),$$

$$f_{11}f_{22} - f_{12}f_{21} + K_1g_2f_{12} + K_2g_1f_{21} - K_1g_1f_{22} - K_2g_2f_{11} = z_1z_2.$$

Із системи з двох рівнянь з двома невідомими неважко знайти коефіцієнти K_1, K_2 (якщо вона сумісна).

Для порівняння знайдемо характеристичне рівняння замкнутого контуру системи (1.10)

$$\det\left(zI_2 - \begin{pmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{pmatrix} + \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} \begin{pmatrix} K_1 & K_2 \end{pmatrix}\right) = z^2 + z(K_1g_1 + K_2g_2 - f_{11} - f_{22}) + f_{11}f_{22} - f_{12}f_{21} + K_1g_2f_{12} + K_2g_1f_{21} - K_1g_1f_{22} - K_2g_2f_{11}.$$

Отже, з урахуванням двох нульових коренів, отримуємо, що абсолютна стійкість когнітивної карти з керуванням забезпечується асимптотичною стійкістю замкнутої системи з регулятором стану. Таким чином, можна обирати коефіцієнти посилення K_1, K_2 так, щоб когнітивна карта була стійкою, що еквівалентно синтезу регулятора по заданому розміщенню полюсів замкненою системою [9].

В загальному випадку, нехай замкнута система з регулятором по стану має вигляд

$$X(k+1) = FX(k) + GU(k), Y(k) = CX(k), U(k) = -KX(k). \quad (1.14)$$

Теорема 3. Динаміка замкненої системи в просторі станів (1.14) може бути представлена у вигляді когнітивної карти з ваговою матрицею суміжності W , де

$$W^T = \begin{pmatrix} -KG & -KF & 0 \\ G & F & 0 \\ CG & CF & 0 \end{pmatrix}. \quad (1.15)$$

Доведення. Рівняння стану і спостерігання еквівалентно перетворюються до виду (1.9). Для перетворення рівняння регулятора розглянемо його на один крок вперед і, підставив в нього праву частину (1.9), отримаємо

$$\begin{aligned}
 \Delta X(k+1) &= F\Delta X(k) + G\Delta U(k), \\
 \Delta Y(k+1) &= CF\Delta X(k) + CG\Delta U(k), \\
 \Delta U(k+1) &= -KF\Delta X(k) - KG\Delta U(k), \\
 X(0) &= X_0, \\
 Y(0) &= Y_0 = CX_0, \\
 U(0) &= -KX_0, \\
 \Delta X(1) &= (F - I)X_0 + GU(0), \Delta U(1) = -K\Delta X(1), \\
 \Delta Y(1) &= C\Delta X(1).
 \end{aligned} \tag{1.16}$$

Ці умови необхідні для еквівалентності динаміки (1.16) і (1.14).

Отже, введення в модель в просторі станів (1.7) керуючої дії веде до появи додаткових ребер в когнітивній карті. А саме: від вершин стану до вершин керування йдуть ребра з вагами, обумовленими коефіцієнтами матриці KF , а вершини керування поєднуються ребрами з коефіцієнтами матриці KG .

$$\begin{pmatrix} \Delta U(k+1) \\ \Delta X(k+1) \\ \Delta Y(k+1) \end{pmatrix} = \begin{pmatrix} -KG & -KF & 0 \\ G & F & 0 \\ CG & CF & 0 \end{pmatrix} \begin{pmatrix} \Delta U(k) \\ \Delta X(k) \\ \Delta Y(k) \end{pmatrix}.$$

На підставі (1.2) слідує, що матриця суміжності задовольняє (1.15). Теорему доведено.

Наслідок 2. Якщо власні числа матриці $F - GK$ по модулю менше одиниці (тобто замкнена система (1.14) асимптотично стійка), то когнітивна модель, задана матрицею суміжності (1.15), буде абсолютно стійкою.

Доведення. Як відомо із теорії керування, модель в просторі станів (1.14) буде асимптотично стійкою, якщо власні числа матриці $F - GK$ по модулю менше одиниці, оскільки, підставив закон керування в рівняння стану, отримаємо рівняння замкненої системи $X(k + 1) = (F - GK)X(k)$. Якщо система (1.14) стійка, всі її змінні будуть обмежені. З іншої сторони, згідно теореми 3, когнітивна модель (1.15) має в точності ту ж динаміку, що й модель в просторі станів (1.14), тобто всі її змінні також будуть обмежені, а це і є визначення абсолютної стійкості когнітивної карти.

Наслідок 3. Якщо систему в просторі станів (1.7) можна стабілізувати, то еквівалентну їй когнітивну модель (1.8) також можна стабілізувати у тому сенсі, що можливо забезпечити її абсолютну стійкість шляхом приведення її до виду (1.15).

Доведення. Систему в просторі станів (1.7) можна стабілізувати, якщо існує така матриця K , при якій замкнена система (1.14) буде стійкою [10]. Оскільки системи (1.14) і (1.15) еквівалентні згідно з теоремою 3, а стійкість (1.14) тягне за собою абсолютну стійкість (1.15) згідно наслідку 2, то з можливості стабілізувати (1.14) випливає можливість стабілізувати (1.15).

1.4 Керування в когнітивній моделі зі спостерігачем стану

Для синтезу керування наведеним вище способом необхідно, щоб усі вершини стану були вимірювані. Це не завжди так. В теорії керування у випадку неспостережуваного вектору стану вводять спостерігач Люенбергера [9], а для керування використовують оцінені за його допомогою значення. Покажемо, яким чином цю методику можна застосувати у когнітивному моделюванні [11].

Розглянемо систему зі спостерігачем і регулятором в просторі станів

$$\begin{aligned} X(k+1) &= FX(k) + GU(k), \\ Y(k) &= CX(k), \end{aligned} \quad (1.17)$$

$$\hat{X}(k+1) = F\hat{X}(k) + GU(k) + L(Y(k) - C\hat{X}(k)), U(k) = -K\hat{X}(k),$$

де $\hat{X}(k)$ – вектор оцінок стану;

L – матриця спостерігача Люенбергера.

Матриця Люенбергера обирається із міркувань стійкості контуру оцінювання, а саме таким чином, щоб корені полінома $\det(F - LC)$ були по модулю менше за одиницю. Перетворимо (1.17) до наступного вигляду

$$\begin{aligned} X(k+1) &= FX(k) + GU(k), \\ Y(k+1) &= CFX(k) + CGU(k), \\ \hat{X}(k+1) &= (F - LC)\hat{X}(k) + GU(k) + LY(k), \\ U(k+1) &= -K(F - LC)\hat{X}(k) - KGU(k) - KLY(k). \end{aligned} \quad (1.18)$$

Для того, щоб представити (1.18) у вигляді когнітивної карти, необхідно ввести додаткові вершини – вершини оцінок $\hat{X}(k)$. Це збільшує розмірність карти, але дозволяє сформувати зворотні зв'язки за керуванням у випадку коли не можливо виміряти вершини стану. Переходячи, як звичайно, до приростів змінних, з урахуванням відповідного вибору початкових значень для оцінок і керувань, в підсумку отримаємо:

$$\begin{pmatrix} \Delta U(k+1) \\ \Delta X(k+1) \\ \Delta \hat{X}(k+1) \\ \Delta Y(k+1) \end{pmatrix} = \begin{pmatrix} -KG & 0 & -K(F - LC) & -KL \\ G & F & 0 & 0 \\ G & 0 & F - LC & L \\ CG & CF & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta U(k) \\ \Delta X(k) \\ \Delta \hat{X}(k) \\ \Delta Y(k) \end{pmatrix}.$$

Отже, в даному випадку

$$W^T = \begin{pmatrix} -KG & 0 & -K(F - LC) & -KL \\ G & F & 0 & 0 \\ G & 0 & F - LC & L \\ CG & CF & 0 & 0 \end{pmatrix}. \quad (1.19)$$

1.5 Перехід від когнітивної моделі до моделі в просторі станів

В більшості практичних ситуацій, коли процес описується когнітивною моделлю (на основі когнітивної карти), для дослідження імпульсних процесів необхідно виконати перехід до моделі у просторі станів. Це дозволяє застосувати для рішення задачі стабілізації імпульсного процесу в когнітивній моделі відомі методи сучасної теорії керування, після чого перейти назад до когнітивної карти за допомогою вищеописаних методів.

Розглянемо певний клас когнітивних карт, для котрих такий перехід легко здійснити. Нехай серед множини вершин Z можна виділити непусту підмножину вершин U , які не мають ребер що в них входять (є витоками) і, при цьому, з точки зору предметної області, можуть розглядатися як такі, якими можна керувати, тобто значення котрих можуть змінюватись обличчям, що приймає рішення. Це потребує втручання людини – експерта, що, до речі, є нормальною практикою в когнітивному моделюванні, оскільки сама когнітивна карта також будується за участі експерта. Якщо така підмножина U існує, задачу переходу від когнітивної карти до моделі в просторі станів можна вирішити. Якщо такої підмножини не існує, можна побудувати модель вільного руху без керуючої дії (типу $X(k + 1) = FX(k)$), однак в цілях стабілізації системи така модель не має користі.

Отже, нехай U існує. Позначимо інші вершини через X і впорядкуємо вершини так, щоб на початку йшли вершини із U (назвемо їх керуючими), потім із X (назвемо їх вершинами стану). Оскільки керуючі вершини є витоками, при

такому впорядкуванні матрицю суміжності можна записати в наступному блочному виді

$$W^T = \begin{pmatrix} 0 & 0 \\ G & F \end{pmatrix}. \quad (1.20)$$

Тоді на основі (1.2) динаміку процесу можна представити як

$$\Delta X(k+1) = F\Delta X(k) + G\Delta U(k). \quad (1.21)$$

Оскільки перехід від моделі в приростах до моделі в абсолютних величинах не є однозначним, ми не будемо вводити додаткові припущення о початкових значеннях, а замість цього запропонуємо, що прирости у вершинах моделі самі по собі являються змінними в просторі станів. Це доволі природньо, якщо враховувати, що звичайно в когнітивному моделюванні розглядаються саме імпульси, тобто прирости, а не абсолютні величини, і тому керувати безпосередньо приростами (як у вершинах керування, так і стану) виявляється зручно. До того ж, у такому випадку не потрібні ті додаткові обмеження на початкові значення, котрі ми вводили при доведенні минулих теорем, оскільки замість моделей (1.7), (1.14) ми будемо одразу мати відповідні моделі в приростах. Відмітимо, що всі інші викладки із минулих підрозділів залишаються майже без змін, тому не будемо їх перевіряти.

Отже, (1.21) – є рівняння стану, що описує когнітивну модель з матрицею суміжності (1.20). Рівняння виміру потрібно не завжди. Зокрема, якщо ми будуємо рівняння за станом на основі теореми 3, то рівняння виміру непотрібне. Легко переконатись, що його, як і відповідні частини матриці суміжності графа, можна просто видалити із доведення теореми і її наслідків. Але у випадку неспостережуваності частини вершин стану для керування необхідно

застосувати спостерігач Люенбергера і теорему 4, для чого рівняння виміру необхідне. Як і у випадку з вибором вершин керування, тут необхідне втручання людини – експерта, котрий встановить, які саме вершини із множини X являються спостережуваними. Позначим їх підмножину через Y (міцністю n_y) і перенумеруємо вершини так, щоби вони слідували у векторі X на початку (для зручності). Тоді з врахуванням (1.21) повна модель в просторі станів буде мати вигляд

$$\Delta X(k+1) = F\Delta X(k) + G\Delta U(k), \quad \Delta Y(k) = C\Delta X(k), \quad (1.22)$$

де $C = \begin{pmatrix} I_{n_y} & 0 \end{pmatrix}$;

0 – нульова матриця розмірності $n_y \times (n_x - n_y)$.

Якщо використовувати рівняння (1.22), отримані описаним чином, в якості перших двох рівнянь системи (1.17) (ототожнивши ΔX з X , ΔY з Y , ΔU з U), можна побудувати регулятор зі спостерігачем за описаним способом й перейти назад до когнітивної моделі. При цьому, окрім додаткових вершин оцінок \hat{X} , формально з'являться нові вершини, що відповідають штучно введеним змінним Y . Цього можна запобігти, перебудувавши модель для випадку матриці C спеціального вигляду, а також застосувавши спостерігач пониженого порядку замість повного. Важливо помітити, що запропонований метод дозволяє перейти від когнітивної моделі до моделі в просторі станів і навпаки, застосувавши до системи один із методів синтезу регулятора стану.

Теорема 4. Когнітивна карта з матрицею суміжності (1.20) може бути еквівалентно представлена моделлю (1.22) в просторі станів. При цьому власні числа матриці F в моделі (1.22) по модулю менше одиниці (система (1.22) асимптотично стійка), то відповідна когнітивна модель, що визначається матрицею суміжності (1.20), буде імпульсно стійкою.

Доведення. Вище вже було доведено еквівалентність динаміки систем (1.22) і (1.20). Із стійкості системи (1.22) випливає, що всі її змінні обмежені. За побудовою змінні (1.22) являються імпульсами (приростами) в когнітивній моделі з матрицею суміжності (1.20), а обмеженість послідовності імпульсів являється визначенням імпульсної стійкості когнітивної карти, що завершує доведення.

За аналогією з теоремою 2 і її наслідками також доводиться, що якщо в систему (1.21) або (1.22), що стабілізується, ввести керування $\Delta U(k) = -K\Delta X(k)$, то отримана когнітивна карта (без врахування несуттєвих тут вершин виміру) буде мати матрицю суміжності

$$W^T = \begin{pmatrix} -KG & -KF \\ G & F \end{pmatrix}. \quad (1.23)$$

В такому випадку імпульсна стійкість отриманої когнітивної карти гарантується тим, що власні числа матриці $F - GK$ по модулю менше одиниці.

1.6 Висновки до розділу 1

В даному розділі було показано як когнітивна модель, що представляє собою орієнтований граф, співвідноситься до моделі у просторі станів, та навпаки, як з моделі у просторі станів (з регулятором, або зі спостерігачем) можна отримати когнітивну модель.

РОЗДІЛ 2

СИСТЕМИ СТАБІЛІЗАЦІЇ. КОГНІТИВНА КАРТА СОЦІАЛЬНО-НАВЧАЛЬНОГО ПРОЦЕСУ СТУДЕНТА

2.1 Когнітивна карта соціально-навчального процесу студента

Покладемо такі вершини когнітивної карти соціально-навчального процесу студента:

- 1) навчання;
- 2) робота;
- 3) сім'я;
- 4) відпочинок;
- 5) здоров'я;
- 6) фінанси.

У когнітивному моделюванні прийнято зводити всі координати вершин когнітивної карти, що відображають фактори різної фізичної природи, до єдиної шкали. Дійсно, оскільки буде будуватись зважений орієнтований граф, експерти не зможуть коректно виставити ваги ребер між вершинами, якщо вони вимірюватимуться в різних одиницях. Тому пропонується запровадити десятибальну шкалу для вимірювання усіх вершин когнітивної карти, де 0 балів — відсутність певного фактору (слабке здоров'я, немає грошей), а 10 — максимальний можливий рівень цього фактору (лише відмінні оцінки за навчання, весь денний час виділяється на роботу тощо). Очевидно, що при встановленні значення фактору на цій шкалі виникає певний суб'єктивізм, який, утім, не заважає моделювати та керувати поведінкою системи в цілому, оскільки головною метою когнітивного моделювання є не отримання точних числових значень, а виявлення тенденцій.

Експертним шляхом було виявлено такі логічні взаємозв'язки між вершинами цієї когнітивної карти:

1. збільшення часу витраченого на навчання призводить до збільшення успіхів у навчанні, але тоді зменшується час що виділяється на роботу, також студент менше відпочиває, його здоров'я зазнає незначного погіршення;

2. коли студент починає більше приділяти увагу роботі, тоді йому доводиться менше відпочивати, це погіршує стан здоров'я, навчальний процес страждає, але успіхи у роботі зазнають покращення, його фінансове становище поліпшується;

3. перед родиною у студента є обов'язки, інколи вони можуть призвести до зменшення часу, що виділяється на навчання, але родина забезпечує студента грошима;

4. збільшення часу що виділяється на відпочинок, призводить до зменшення часу, що виділяється на навчання та роботу, але здоров'я студента поліпшується. На відпочинок студент витрачає кошти;

5. покращення стану здоров'я студента, призводить до поліпшення його навчального та робочого процесу, поліпшується сімейне благополуччя. Студент може менше відпочивати, та витратити менше грошей на підтримку здоров'я;

6. збільшення фінансового благополуччя дозволяє студенту більше відпочивати.

Для моделювання було побудовано когнітивну карту у вигляді зваженого орграфа (рис. 2.1), на якому ваги ребер чисельно відображають ступінь впливу кожної вершини на інші.

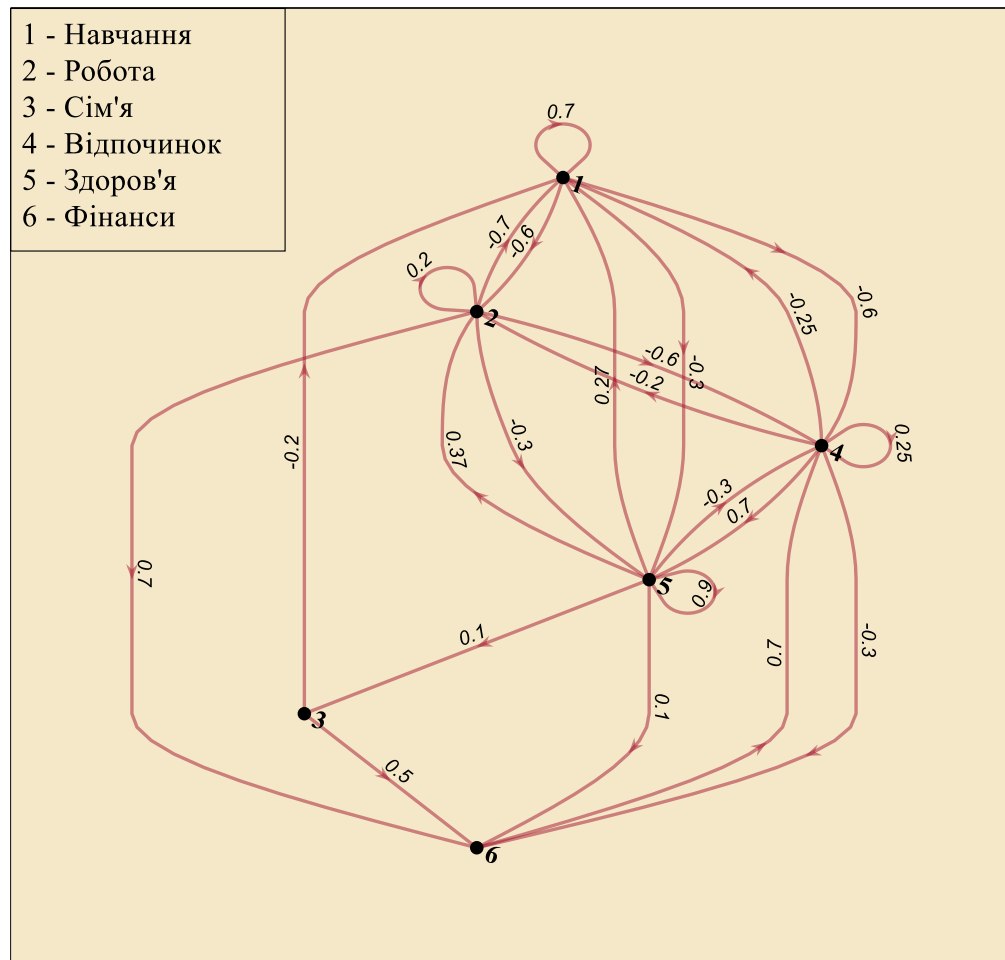


Рисунок 2.1

Імпульсний процес у системі, що зображується цією моделлю, має динаміку, яка описується рівнянням

$$\Delta Y(k+1) = A \Delta Y(k), \quad (2.1)$$

або

$$\begin{pmatrix} \Delta Y_1(k+1) \\ \Delta Y_2(k+1) \\ \Delta Y_3(k+1) \\ \Delta Y_4(k+1) \\ \Delta Y_5(k+1) \\ \Delta Y_6(k+1) \end{pmatrix} = \begin{pmatrix} 0.7 & -0.6 & 0 & -0.6 & -0.3 & 0 \\ -0.7 & 0.2 & 0 & -0.6 & -0.3 & 0.7 \\ -0.2 & 0 & 0 & 0 & 0 & 0.5 \\ -0.25 & -0.2 & 0 & 2.5 & 0.7 & -0.3 \\ 0.27 & 0.37 & 0.1 & -0.3 & 0.9 & 0.1 \\ 0 & 0 & 0 & 0.7 & 0 & 0 \end{pmatrix}^T \begin{pmatrix} \Delta Y_1(k) \\ \Delta Y_2(k) \\ \Delta Y_3(k) \\ \Delta Y_4(k) \\ \Delta Y_5(k) \\ \Delta Y_6(k) \end{pmatrix}.$$

де $\Delta Y(k) = Y(k) - Y(k - 1)$;

A – вагова матриця суміжності.

Нескладно перевірити, що система є нестійкою, тому що серед власних чисел матриці A є за модулем більші за одиницю. А саме, власні числа системи

$$\begin{aligned} z_1 &= -0.6753, \\ z_2 &= 1.1702, \\ z_3 &= 0.5309 + 0.7413i, \\ z_4 &= 0.5309 - 0.7413i, \\ z_5 &= 0.5174, \\ z_6 &= -0.0241, \end{aligned}$$

Це означає, що після будь-якого, навіть незначного, зовнішнього збурення, коли система перейде в режим імпульсного процесу, якщо не здійснювати ніяких додаткових дій, стан студента почне різко змінюватись по всіх “координатах” одночасно, що може призвести до вкрай негативних наслідків. Це можна спостерігати на прикладі студентів, у яких після тривалого усталеного стану раптово починаються проблеми у навчанні, роботі, сім’ї тощо.

2.2 Розробка системи стабілізації нестійкого імпульсного процесу когнітивної карти в приростах змінних

Якщо використовувати оператор зворотного зсуву q^{-1} . Тоді модель вільного руху імпульсного процесу (2.1) можна зобразити у вигляді [12]

$$\Delta Y_i(k) = \sum_{j=1}^n a_{ij} \Delta Y_j(k - 1) = \sum_{j=1}^n a_{ij} q^{-1} \Delta Y_j(k), \quad i = 1, \dots, n. \quad (2.2)$$

Напишемо цей вираз в розгорнутій формі

$$\begin{pmatrix} 1 - a_{11}q^{-1} & -a_{12}q^{-1} & \dots & -a_{1n}q^{-1} \\ -a_{21}q^{-1} & 1 - a_{22}q^{-1} & \dots & -a_{2n}q^{-1} \\ \dots & \dots & \dots & \dots \\ -a_{n1}q^{-1} & -a_{n2}q^{-1} & \dots & 1 - a_{nn}q^{-1} \end{pmatrix} \begin{pmatrix} \Delta Y_1(k) \\ \Delta Y_2(k) \\ \dots \\ \Delta Y_n(k) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \end{pmatrix}. \quad (2.3)$$

При введенні матричного полінома $A(q^{-1})$ відносно оператора зворотного зсуву

$$A(q^{-1}) = \begin{pmatrix} 1 - a_{11}q^{-1} & -a_{12}q^{-1} & \dots & -a_{1n}q^{-1} \\ -a_{21}q^{-1} & 1 - a_{22}q^{-1} & \dots & -a_{2n}q^{-1} \\ \dots & \dots & \dots & \dots \\ -a_{n1}q^{-1} & -a_{n2}q^{-1} & \dots & 1 - a_{nn}q^{-1} \end{pmatrix}, \quad (2.4)$$

вираз (2.3) можна записати як

$$A(q^{-1})\Delta Y(k) = 0. \quad (2.5)$$

Для розробки системи керування імпульсним процесом когнітивної карти, будемо вважати всі вершини керованими, та доповнимо модель вільного руху (2.5) впливом зовнішніх вимушених сил у вигляді вектору приростів керувань $\Delta u(k) = u(k) - u(k-1)$, котрі взаємодіють безпосередньо на вершини когнітивної карти. В результаті модель керованого імпульсного процесу когнітивної карти типу «вхід-вихід» можна зобразити у вигляді

$$(I - A_1q^{-1})\Delta Y(k) = B(q^{-1})\Delta u(k), \quad (2.6)$$

де $I - A_1q^{-1} = A(q^{-1})$;

$B(q^{-1}) = B_1q^{-1}$ – матриця керування.

При цьому для формування зовнішніх керувань, які будуть впливати на вершини когнітивної карти, діагональну матрицю B_1 в загальному випадку можна обрати в формі одиничної матриці $B_1 = I$. Згідно з (2.2), (2.3), (2.4) матриця A_1 , складена із коефіцієнтів моделі (2.1), та матриця B_1 мають вид

$$\begin{aligned} A_1 &= \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}, \\ B_1 &= \begin{pmatrix} b_{11} & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & b_{nn} \end{pmatrix}. \end{aligned} \quad (2.7)$$

Згідно з (2.6), (2.7) приріст кожної координати вершини когнітивної карти $Y_i(k)$ буде визначатись як

$$\Delta Y_i(k) = a_{ii}\Delta Y_i(k-1) + \sum_{j=1, j \neq i}^n a_{ij}\Delta Y_j(k-1) + b_{ii}\Delta u_i(k-1), \quad (i = 1, \dots, n).$$

Для моделі керованого імпульсного процесу (2.6) сформулюємо закон дискретного керування в приростах змінних в загальному вигляді [7]

$$\Delta u(k) = D(q^{-1})(I + F(q^{-1}))^{-1}(\Delta G(k) - \Delta Y(k)). \quad (2.8)$$

де $\Delta G(k)$ – заданий вектор приростів впливів, до якого має прагнути вектор приростів координат вершин когнітивної карти $\Delta Y(k)$ (на практиці задаються $\Delta G(k) = 0$).

Для моделі (2.6) структура складових закону керування (2.8) буде наступною

$$D(q^{-1}) = D_0, \quad F(q^{-1}) = F_1 q^{-1}. \quad (2.9)$$

Підставимо (2.8) у (2.6). Тоді після перетворень отримаємо рівняння багатовимірної замкненої системи керування по каналу $\Delta G \rightarrow \Delta Y$:

$$\begin{aligned} \Delta Y(k) = (I - F_1 q^{-1}) \{ (I - A_1 q^{-1})(I + F_1 q^{-1}) + q^{-1} B_1 D_0 \}^{-1} B_1 q^{-1} D_0 \times \\ \times (I + F_1 q^{-1})^{-1} \Delta G(k). \end{aligned} \quad (2.10)$$

З якого виділимо характеристичне рівняння

$$\det((I - A_1 q^{-1})(I + F_1 q^{-1}) + q^{-1} B_1 D_0) = 0. \quad (2.11)$$

Для забезпечення стійкості замкненої системи (2.10) при нестійкому імпульсному процесі когнітивної карти (2.2), (2.5) сформуємо еталонну модель характеристичного поліному дискретної передаточної функції замкненої системи (2.10)

$$I + A_m(q^{-1}) = (I - A_1 q^{-1})(I + F_1 q^{-1}) + q^{-1} B_1 D_0. \quad (2.12)$$

При цьому корені еталонної моделі $\det(I + A_m(q^{-1})) = 0$ обираються по модулю менше одиниці. Для забезпечення рівності (2.12) порядок еталонної моделі має бути рівним двом, тобто

$$I + A_{m1} q^{-1} + A_{m2} q^{-2} = (I - A_1 q^{-1})(I + F_1 q^{-1}) + q^{-1} B_1 D_0. \quad (2.13)$$

Після перемноження і приведення подібних членів в правій частині, прирівняв коефіцієнти при відповідних степенях, отримаємо рівності

$$A_{m1} = -A_1 + F_1 + B_1 D_0,$$

$$A_{m2} = -A_1 F_1.$$

Із цих рівнянь визначимо параметри закону керування (2.9)

$$F_1 = -A_1^{-1} A_{m2}, \quad (2.14)$$

$$D_0 = B_1^{-1} (A_1 + A_1^{-1} A_{m2} + A_{m1}). \quad (2.15)$$

Таким чином, на кожному кроці дискретизації, при вимірних приростах координат вершин $\Delta Y(k)$ на основі закону керування (2.8) і параметрів регулятора (2.14), (2.15), визначається приріст $\Delta u(k)$ (при $\Delta G(k) = 0$)

$$\Delta u(k) = B_1^{-1} (A_1 + A_1^{-1} A_{m2} + A_{m1}) (I - A_1^{-1} A_{m2} q^{-1})^{-1} \Delta Y(k).$$

В явному вигляді після перетворень приріст керуючої дії можна записати так

$$\begin{aligned} \Delta u(k) = B_1^{-1} (A_1 + A_1^{-1} A_{m2} + A_{m1}) (A_1^{-1} A_{m2} (A_1 + A_1^{-1} A_{m2} + A_{m1})^{-1} \times \\ \times B_1 \Delta u(k-1) - \Delta Y(k)). \end{aligned} \quad (2.16)$$

Для того щоб перейти від замкненої системи керування (2.6), (2.16) назад до когнітивної карти, необхідно представити її в наступному вигляді

$$\Delta Y(k) = A_1 \Delta Y(k-1) + B_1 \Delta u(k-1),$$

$$\begin{aligned} \Delta u(k) = B_1^{-1} (A_1 + A_1^{-1} A_{m2} + A_{m1}) (A_1^{-1} A_{m2} (A_1 + A_1^{-1} A_{m2} + A_{m1})^{-1} \times \\ \times B_1 \Delta u(k-1) - A_1 \Delta Y(k) - B_1 \Delta u(k-1)), \end{aligned}$$

або інакше

$$\begin{aligned}\Delta Y(k) &= A_1 \Delta Y(k-1) + B_1 \Delta u(k-1), \\ \Delta u(k) &= B_1^{-1}(A_1 + A_1^{-1}A_{m2} + A_{m1})(A_1^{-1}A_{m2}(A_1 + A_1^{-1}A_{m2} + A_{m1})^{-1} - I) \times \\ &\quad \times B_1 \Delta u(k-1) - B_1^{-1}(A_1 + A_1^{-1}A_{m2} + A_{m1})A_1 \Delta Y(k-1).\end{aligned}$$

Тоді динаміку нової когнітивної карти можна зобразити так

$$\begin{pmatrix} \Delta Y(k) \\ \Delta u(k) \end{pmatrix} = \begin{pmatrix} A_1 & B_1 \\ -B_1 Z A_1 & B_1^{-1} Z (A_1^{-1} A_{m2} Z^{-1} - I) B_1 \end{pmatrix} \begin{pmatrix} \Delta Y(k-1) \\ \Delta u(k-1) \end{pmatrix},$$

де $Z = A_1 + A_1^{-1}A_{m2} + A_{m1}$.

Таким чином, в загальному випадку матриця суміжності когнітивної карти, відповідної замкненої системи керування, буде мати вигляд

$$\begin{pmatrix} A_1 & -B_1 Z A_1 \\ B_1 & B_1^{-1} Z (A_1^{-1} A_{m2} Z^{-1} - I) B_1 \end{pmatrix},$$

де коефіцієнти A_1 визначаються із початкової когнітивної карти.

2.3 Розробка системи адаптивного керування імпульсним процесом при невідомих і змінних коефіцієнтах когнітивної карти

В багатьох випадках коефіцієнти вагової матриці когнітивної карти не можуть бути визначені точно, оскільки зазвичай вони визначаються шляхом застосування експертних оцінок. Окрім того, в процесі функціонування системи, що описує когнітивна карта, вони можуть змінюватись з часом. Однак описаний

вище алгоритм керування базується на відомих коефіцієнтах моделі системи. У зв'язку з цим виникає задача адаптивного керування когнітивною картою. В процесі функціонування системи має одночасно виконуватись керування, і оцінка параметрів когнітивної карти [12].

Будемо використовувати рекурентний метод найменших квадратів, припускаючи, що збурення, викликані неточним знанням коефіцієнтів моделі, мають характер білого шуму. Напишем рівність (2.6) по координатно

$$\Delta Y_i(k) = \sum_{j=1}^n a_{ij} \Delta Y_j(k-1) + b_i \Delta u_i(k-1) + \xi_i(k). \quad (2.17)$$

Слід враховувати, що, взагалі кажучи, частина коефіцієнтів a_{ij} заздалегідь рівна нулю (у тих випадках, коли між відповідними вершинами нема зв'язку). Нехай ненульовими в i -му рядку матриці A являються коефіцієнти $a_{ij_1}, \dots, a_{ij_{p_i}}$. Позначимо $X_i(k) = [\Delta Y_{j_1}(k-1), \dots, \Delta Y_{j_{p_i}}(k-1)]^T$, $\Theta_i = [a_{ij_1}, \dots, a_{ij_{p_i}}]$. Тоді (2.17) можна записати так

$$\Delta Y_i(k) - b_i \Delta u_i(k-1) = X_i^T(k) \Theta_i + \xi_i(k). \quad (2.18)$$

Позначимо $\Theta_i(k)$ поточну оцінку Θ_i . Застосуємо рекурентний метод найменших квадратів, отримаємо наступний алгоритм оцінки параметрів системи

$$\begin{aligned} \Theta_i(k) &= \Theta_i(k-1) + K_i(k) \left(\Delta Y_i(k) - b_i \Delta u_i(k-1) - X_i^T(k) \Theta_i(k-1) \right), \\ K_i(k) &= \frac{1}{1 + X_i^T(k) P_i(k-1) X_i(k)} P_i(k-1) X_i(k), \\ P_i(k) &= P_i(k-1) - \frac{1}{1 + X_i^T(k) P_i(k-1) X_i(k)} P_i(k-1) X_i(k) X_i^T(k) P_i(k-1). \end{aligned} \quad (2.19)$$

Рекурентну процедуру (2.19) необхідно проводити для кожної вершини карти $\Delta Y_i(k)$ на кожному кроці дискретизації. Отримані оцінки $\Theta_i(k)$ необхідно використовувати замість істинних значень коефіцієнтів матриці A на поточному кроці алгоритму керування (2.16).

2.4 Розробка системи стабілізації координат вершин когнітивної карти на заданих рівнях при нестійкому імпульсному процесі

Розглянемо рівняння вільного руху імпульсного процесу когнітивної карти (2.2), представленого не в приростах, а в повних координатах вершин. Його можна записати в наступному вигляді [12]

$$(I - (I + A_1)q^{-1} + A_1q^{-2})Y(k) = 0. \quad (2.20)$$

При умові, що $\lim_{k \rightarrow \infty} \Delta Y_i(k) = 0$. Для стабілізації координат вершин моделі Y_i на заданих рівнях G_i необхідно формувати зовнішні керуючі дії u_i , які впливають безпосередньо на вершини когнітивної карти по синтезованому закону керування. При цьому допускається, що вільний рух когнітивної карти (2.20) нестійкий, тобто серед коренів рівняння $\det(I - (I + A_1)q^{-1} + A_1q^{-2}) = 0$ є корені по модулю більше за одиницю. Динаміку вимушеного руху вершин карти при впливі зовнішніх керувань можна представити за допомогою моделі

$$(I - (I + A_1)q^{-1} + A_1q^{-2})Y(k) = B_1q^{-1}u(k). \quad (2.21)$$

Закон керування в загальному виді сформуємо наступним чином

$$u(k) = (D_0 + D_1 q^{-1})(I + F_1 q^{-1})^{-1}(G(k) - Y(k)) \quad (2.22)$$

При підстановці (2.22) в (2.21) після перетворень отримаємо рівняння замкненої системи

$$Y(k) = (I + F_1 q^{-1})\{(I - (I + A_1)q^{-1} + A_1 q^{-2})(I + F_1 q^{-1}) + B_1 q^{-1}(D_0 + D_1 q^{-1})\}^{-1} B_1 q^{-1}(D_0 + D_1 q^{-1})(I + F_1 q^{-1})^{-1} G(k). \quad (2.23)$$

Для характеристичного полінома замкненої системи (2.23) сформуємо наступну еталонну модель:

$$\begin{aligned} I + A_{m1}q^{-1} + A_{m2}q^{-2} + A_{m3}q^{-3} = \\ = (I - (I + A_1)q^{-1} + A_1 q^{-2})(I + F_1 q^{-1}) + B_1 q^{-1}(D_0 + D_1 q^{-1}). \end{aligned} \quad (2.24)$$

де корні полінома $\det(I + A_{m1}q^{-1} + A_{m2}q^{-2} + A_{m3}q^{-3})$ обираються по модулю менше одиниці. Після перемноження виразів в правій частині (2.24) і приведення подібних членів, прирівнюючи коефіцієнти при відповідних степенях, отримаємо рівності

$$\begin{aligned} A_{m1} &= F_1 + B_1 D_0 - I - A_1, \\ A_{m2} &= A_1 - (I + A_1)F_1 + B_1 D_1, \\ A_{m3} &= A_1 F_1. \end{aligned}$$

Із них визначаються параметри закону керування (2.22)

$$\begin{aligned} F_1 &= A_1^{-1} A_{m3}, \\ D_0 &= B_1^{-1}(I + A_1 + A_{m1} - A_1^{-1} A_{m3}), \end{aligned}$$

$$D_1 = B_1^{-1}((I + A_1)A_1^{-1}A_{m3} - A_1 + A_{m2}).$$

2.5 Висновки до розділу 2

В даному розділі було побудовано когнітивну модель соціально-навчального процесу студента, та відповідну їй модель у просторі станів. До останньої було синтезовано регулятор стану на базі еталонних моделей, для стабілізації нестійкого імпульсного режиму. На випадок невідомих значень параметрів моделі, було продемонстровано застосування рекурентного методу найменших квадратів, для їх визначення. Також було побудовано систему стабілізації вершин когнітивної карти на заданих рівнях.

\

РОЗДІЛ 3

ЧИСЕЛЬНЕ МОДЕЛЮВАННЯ. АНАЛІЗ РЕЗУЛЬТАТІВ

3.1 Моделювання нестійкого імпульсного режиму соціально-навчального процесу студента

Нехай на початку навчального семестру, що триває 14 тижнів, студент має гарні показники у навчанні, добре справляється з роботою, у сім'ї все гаразд, інколи відпочиває, має гарне здоров'я та достатньо грошей. Цей стан студента можна зобразити початковими координатами $Y_0 = (7; 6; 5; 3; 7; 4)$. Тепер нехай в перший тиждень студент раптово захворів. Цю подію можна відобразити як імпульс $\Delta Y_0 = (0; 0; 0; 0; -3; 0)$, який буде початковими координатами для системи (2.2). Вільний рух системи, що зображено на рисунку 3.1, отримаємо за формулою $Y_i(k+1) = Y_i(k) + \Delta Y(k+1)$. Імпульсний процес (2.2) зображено на рисунку 3.2.

Прокоментуємо отримані результати. В перші тижні хвороби почали страждати успіхи у навчанні і роботі. Фінансове становище зазнає значних втрат. Студент починає багато відпочивати, аби розпочати процес одужання. Коли здоров'я починає відновлюватись (тиждень 4), він вертається до навчання і до роботи. Але на роботу він починає прикладати більше зусиль ніж на навчання, це зумовлено необхідністю збільшення фінансового стану, який до цього моменту зовсім занепав.

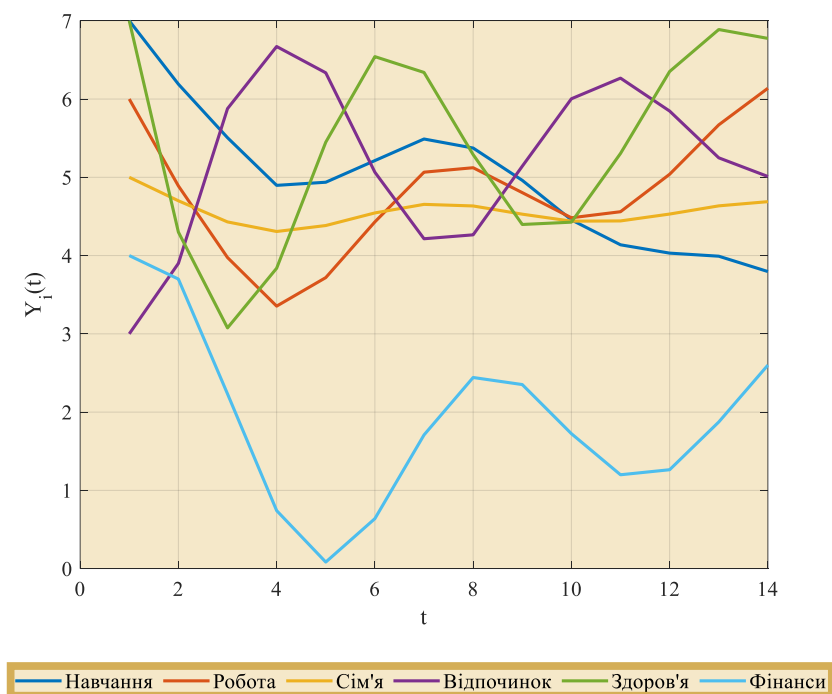


Рисунок 3.1

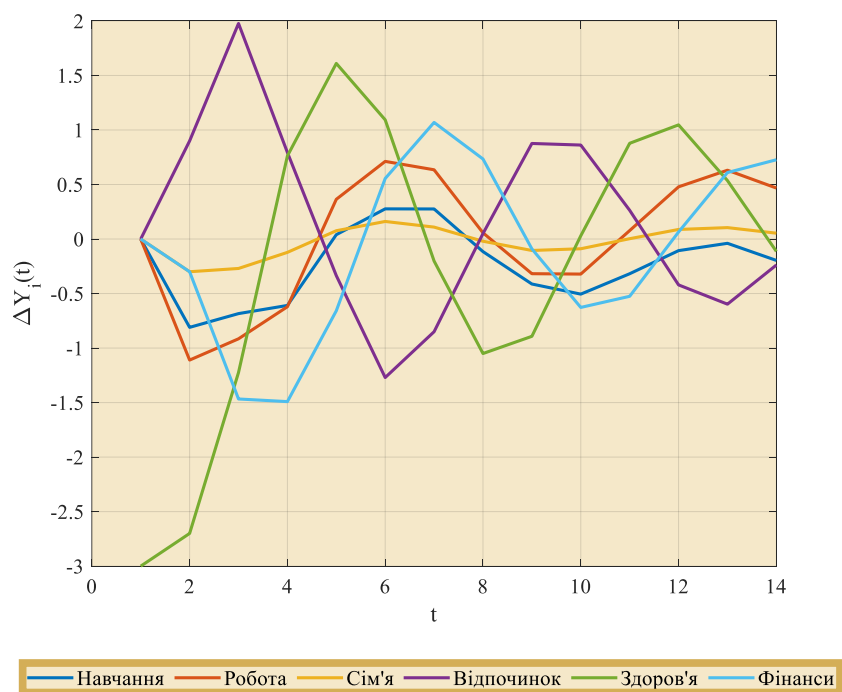


Рисунок 3.2

На цьому тижні, здоров'я знову почало падати, а за ним незначного погіршення зазнали навчальний та робочий процеси, але коли здоров'я знов почало відновлюватись, студент перестав займатись навчанням, усі його зусилля були спрямовані на роботу та поліпшення фінансового становища. Як наслідок на кінець семестру студент має низькі успіхи у навчанні, що призводить до відрахування.

3.2 Система стабілізації соціально-навчального процесу студента

Побудуємо систему керування когнітивною картою за алгоритмами (2.6), (2.16). В якості еталонної моделі замкненої системи візьмемо діагональний матричний поліном з однаковими поліномами на головній діагоналі. Нехай корені характеристичного рівняння дорівнюють $z_1 = -0.6$ і $z_2 = 0.1$, тоді матимемо матриці $A_{m1} = 0.5I$, $A_{m2} = -0.06I$. Початкові данні залишимо такими як і в підрозділі 3.1. Динаміку системи показано на рисунку 3.3, а її імпульсний процес на рисунку 3.4.

Результати моделювання показують, що систему, стабілізовано. По закінченню перехідного процесу стан системи відповідає вектору $Y_c = (6.25; 4.9; 4.7; 3.8; 4.3; 3.7)$. Різниця між сталим станом Y_c і початковим Y_0 дорівнює

$$Y_c - Y_0 = (-0.75; -1.1; -0.3; 0.8; -2.7; -0.3) \quad (3.1)$$

На рисунках 3.5 та 3.6 зображено керуючі дії $U(k)$ та їх імпульси $\Delta U(k)$, відповідно.

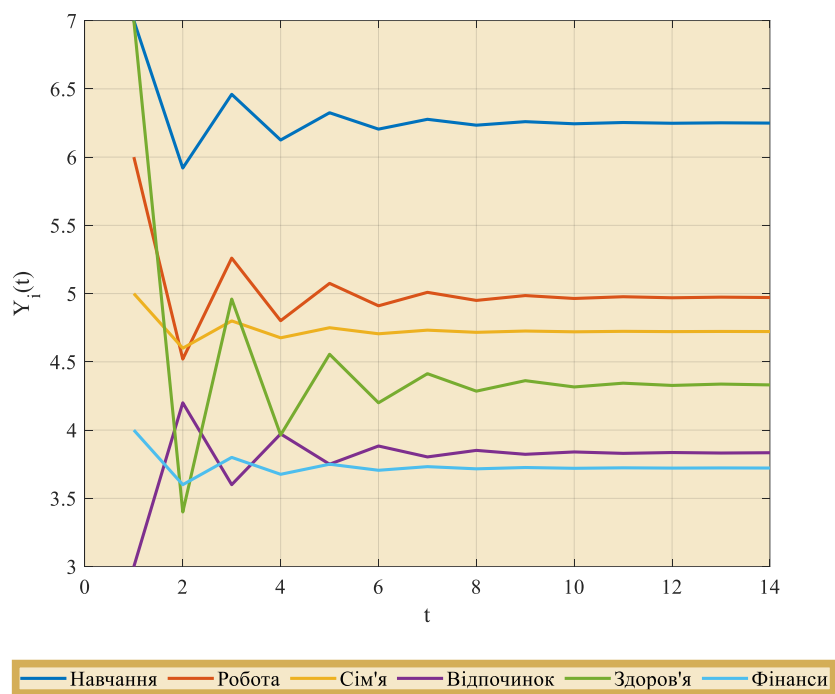


Рисунок 3.3

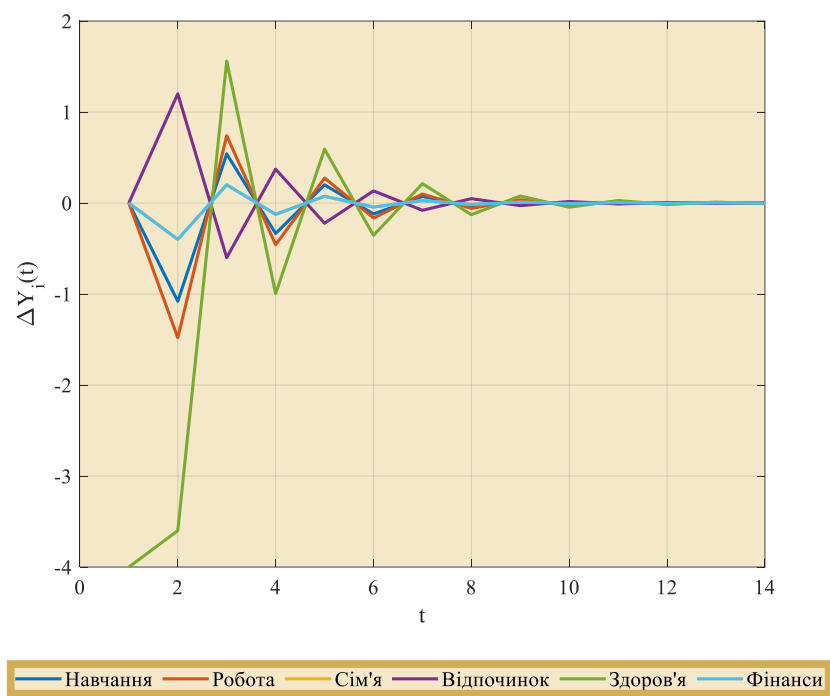


Рисунок 3.4

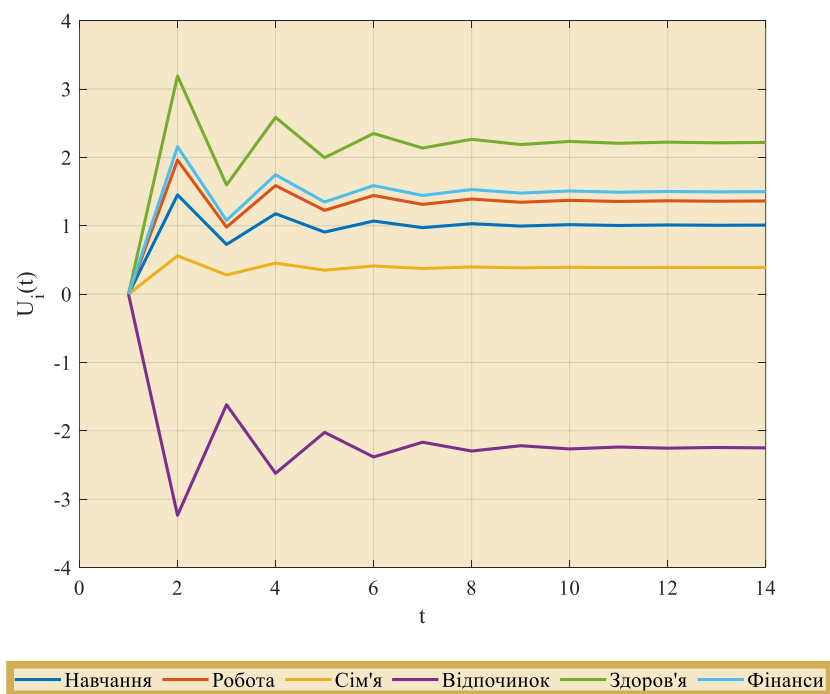


Рисунок 3.5

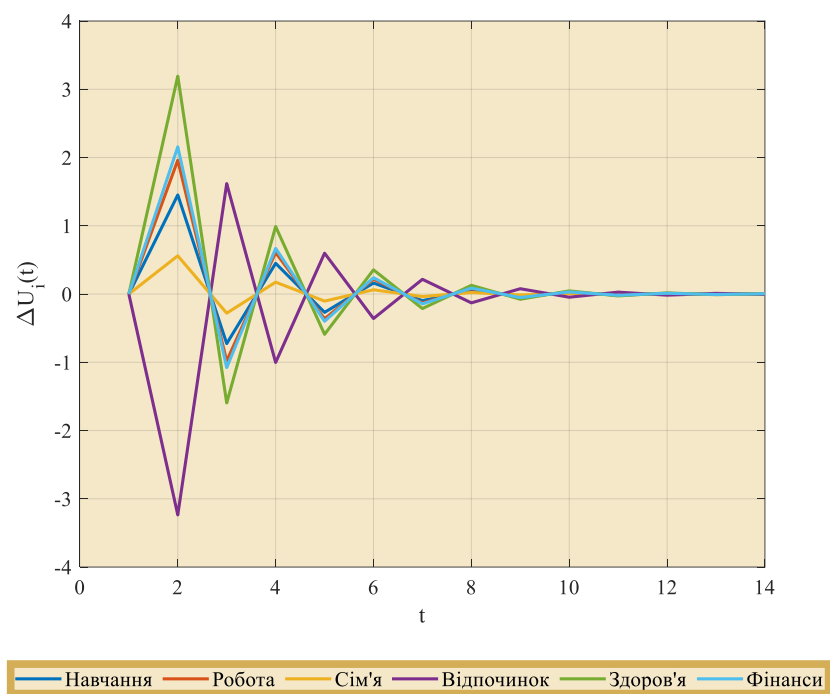


Рисунок 3.6

З (3.1) видно, що стан студента, по закінченню перехідного процесу зазнав негативних змін, окрім координати «відпочинок». Координату «здоров'я» було стабілізовано на рівні нижче середнього, що не є гарно, але на відміну від результатів моделювання підрозділу 3.1, координати «навчання», «робота» та «фінанси» на початок сесії залишаються на досить високому рівні, що дозволяє студенту успішно здати іспити і продовжити навчання у наступному семестрі.

З рисунку 3.6 можна зробити висновки, що студенту для стабілізації свого стану необхідно покращувати свій стан здоров'я, але робити це в меншій ступені за рахунок відпочинку, аби не нашкодити сильно своєму навчальному та робочому процесу.

3.3 Система стабілізації соціально-навчального процесу студента при невідомій матриці суміжності

Тепер припустимо, що ваги когнітивної карти невідомі. Нехай замість них є наступні початкові оцінки

$$Q_0 = \begin{pmatrix} 0.4 & -0.4 & 0 & -0.4 & -0.1 & 0 \\ -0.5 & 0.6 & 0 & -0.9 & -0.5 & 0.9 \\ -0.5 & 0 & 0 & 0 & 0 & 0.2 \\ -0.4 & -0.6 & 0 & 0.1 & 0.4 & -0.6 \\ 0.5 & 0.1 & 0.2 & -0.6 & 0.7 & 0.3 \\ 0 & 0 & 0 & 0.3 & 0 & 0 \end{pmatrix}^T.$$

Скористаємося алгоритмом (2.19) для рекурентного оцінювання невідомих параметрів, та будемо використовувати ці оцінки замість істинних значень у (2.16). Покладемо $P_i(1) = 1000I$ для всіх $i = 1, \dots, 6$. Результати моделювання

показані на рисунках 3.7, 3.8, 3.9 та 3.10. На рисунку 3.11 показано як змінювались оцінки параметрів моделі з часом.

Отримані результати демонструють стабілізацію системи при невідомих параметрах моделі, маючи лише їх оцінки Q_0 . З рисунку 3.11 видно як оцінки параметрів моделі Q вже на шостій ітерації ледь відмінюються від істинних значень параметрів.

Порівнюючи отримані результати з результатами моделювання у минулому підрозділі, помітно як суттєво відрізняються сталі стани моделі. Найбільш помітні зміни у координатах «навчання» та «відпочинок».

Керуючи імпульси (рис. 3.10) привели до того, що здоров'я стабілізувалось, на рівні нижче середнього, відпочивати студент став менше ніж на початку, стан роботи не зазнав змін, а навчання, навпаки, покращилося. Характерною рисою керуючих імпульсів є те, що вони циклічно затухають, до того ж з них видно як чередуються необхідність студенту відпочивати з необхідністю навчатися і працювати.

Стабілізація когнітивної карти відбулася, але отримані результати можуть не задовольняти, адже стан здоров'я студента було стабілізовано на низькому рівні, що доволі погано. Для вирішення цієї проблеми в підрозділі 2.3 було розроблено систему стабілізації вершин когнітивної карти на заданих рівнях G . Наступний підрозділ присвячено застосуванню даної системи.

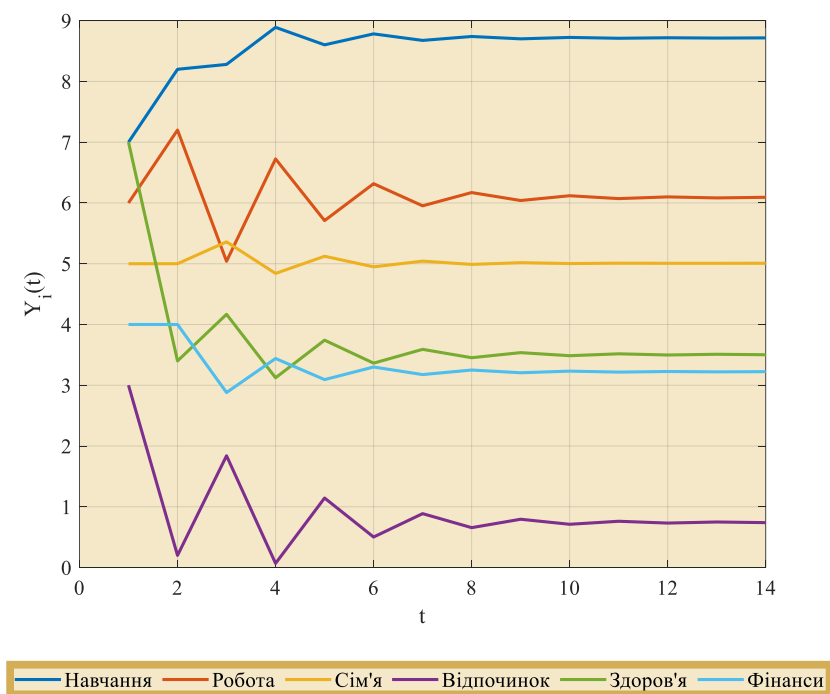


Рисунок 3.7

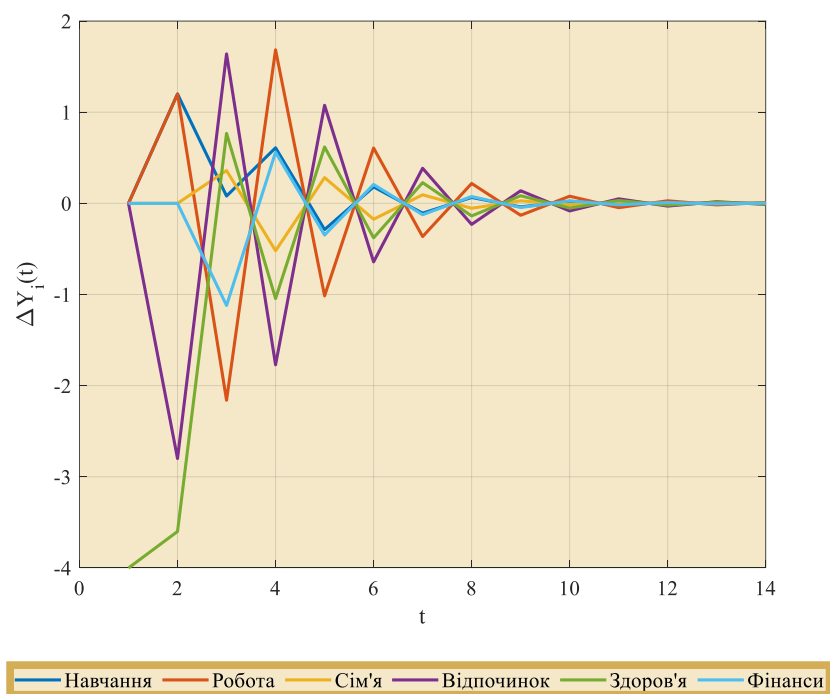


Рисунок 3.8

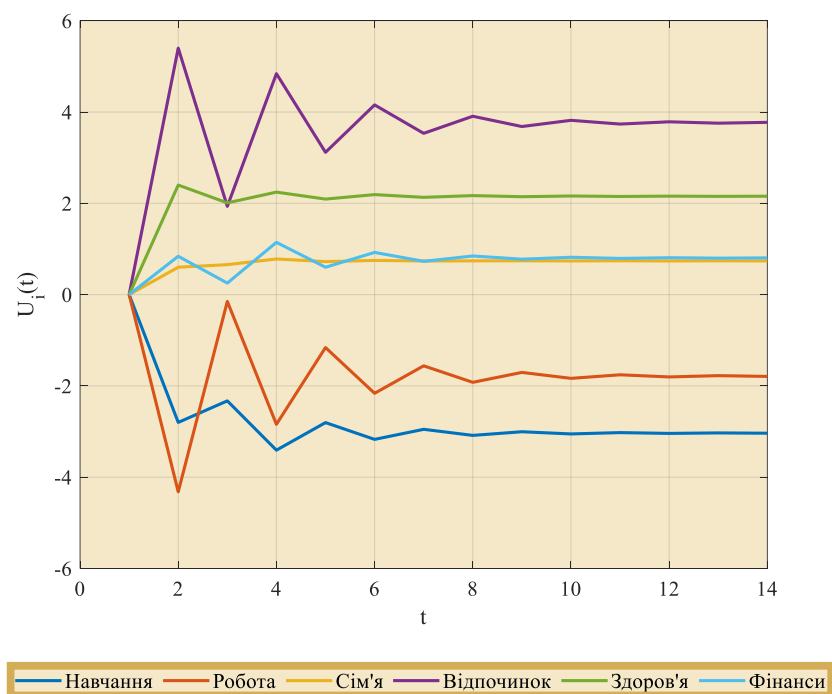


Рисунок 3.9

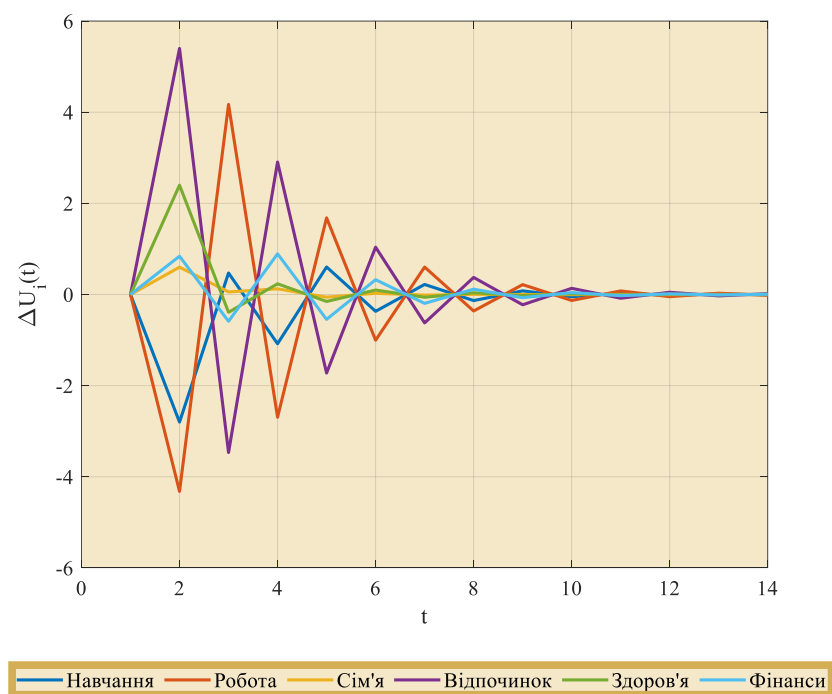


Рисунок 3.10

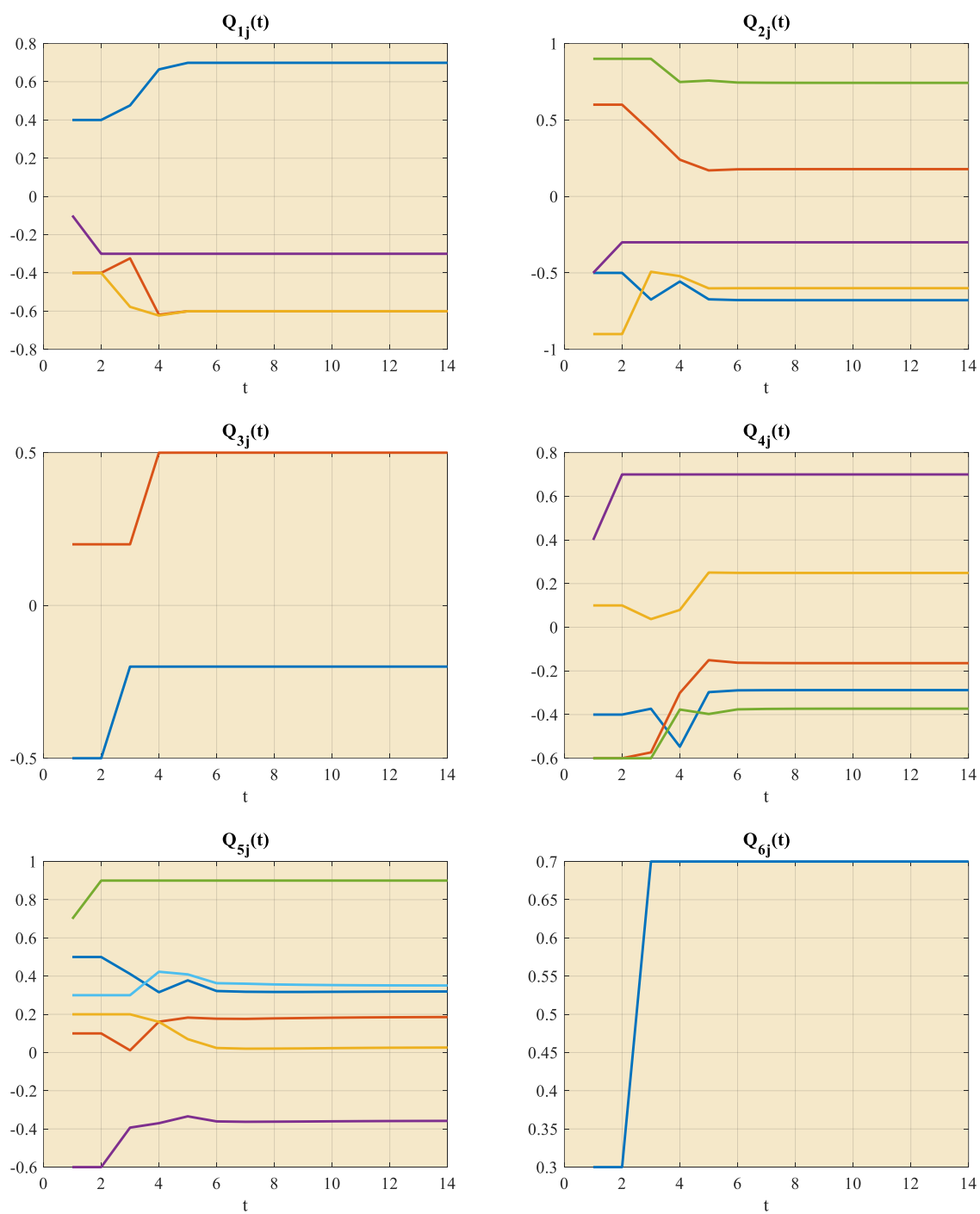


Рисунок 3.11

3.4 Система стабілізації соціально-навчального процесу студента на заданих рівнях

Нехай студент, що раптово захворів, хоче вернутися до свого початкового стану $Y_0 = (7; 6; 5; 3; 7; 4)$, тобто стабілізувати вершини когнітивної карти на рівні $G = (7; 6; 5; 3; 7; 4)$. Виконаємо чисельне моделювання за формулами (2.21), (2.22). В якості еталонної моделі замкненої системи візьмемо діагональний матричний поліном з однаковими поліномами на головній діагоналі. Нехай корні характеристичного рівняння будуть рівні $-0.3, 0.2, -0.4$. Тоді будемо мати такі коефіцієнти $A_{m1} = 0.5I$, $A_{m2} = -0.02I$, $A_{m3} = -0.024I$. Результати моделювання показано на рисунках 3.12, 3.13 та 3.14.

Стан студента було відновлено в більшості за рахунок лікування, в процесі якого студент не тільки відпочивав, а й виділяв час на навчання та, по можливості, виконував свої робочі обов'язки. Динаміка керуючих дій, що зображена на рисунку 3.13, має циклічний затухаючий характер, в якому відпочинок чередується з навчанням, працею та доглядом за здоров'ям.

Результати моделювання, що було отримано в цьому розділі не суперечать здоровому глузду. Це показує адекватність побудованої когнітивної моделі соціально-навчального процесу студента, та можливість застосовувати до неї методи теорії керування.

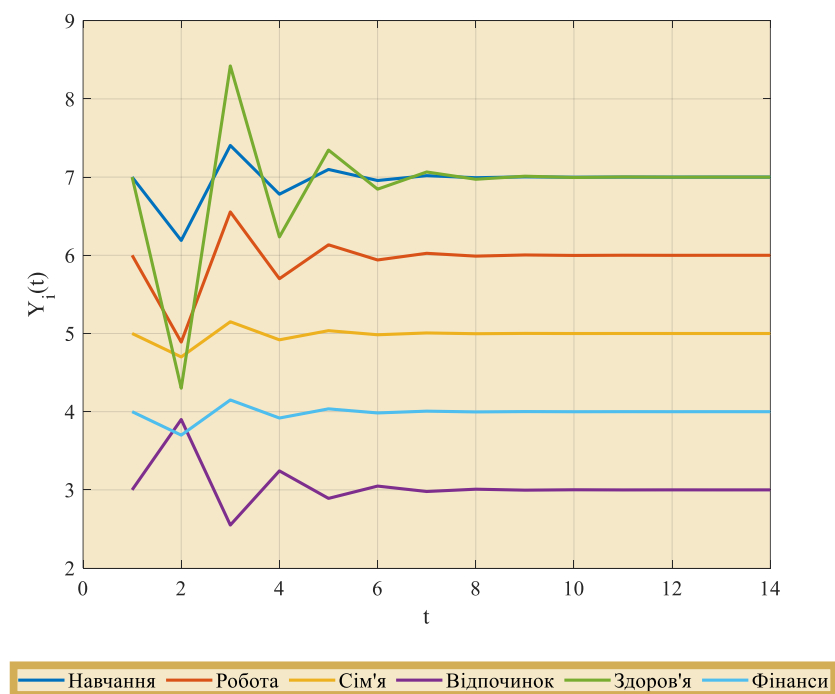


Рисунок 3.12

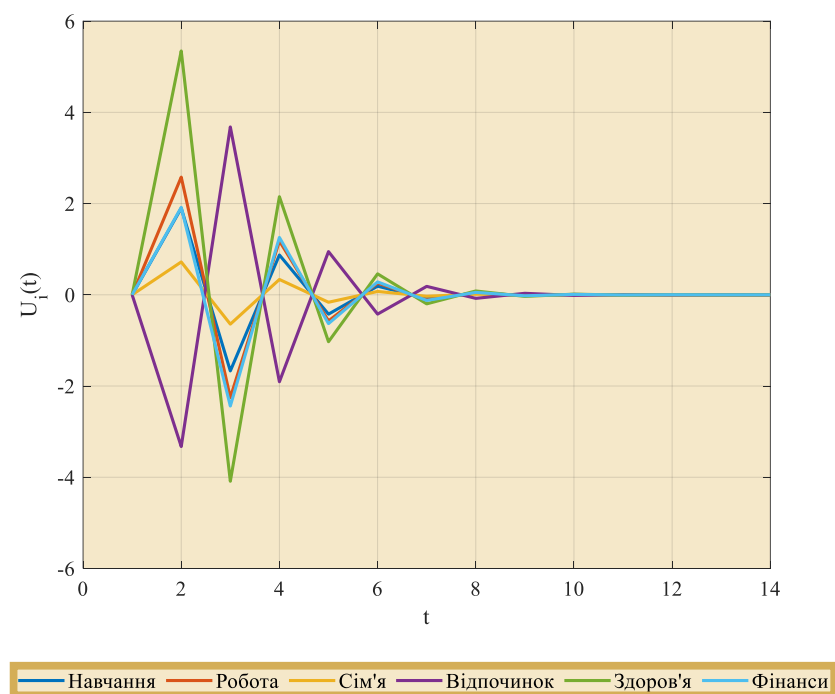


Рисунок 3.13

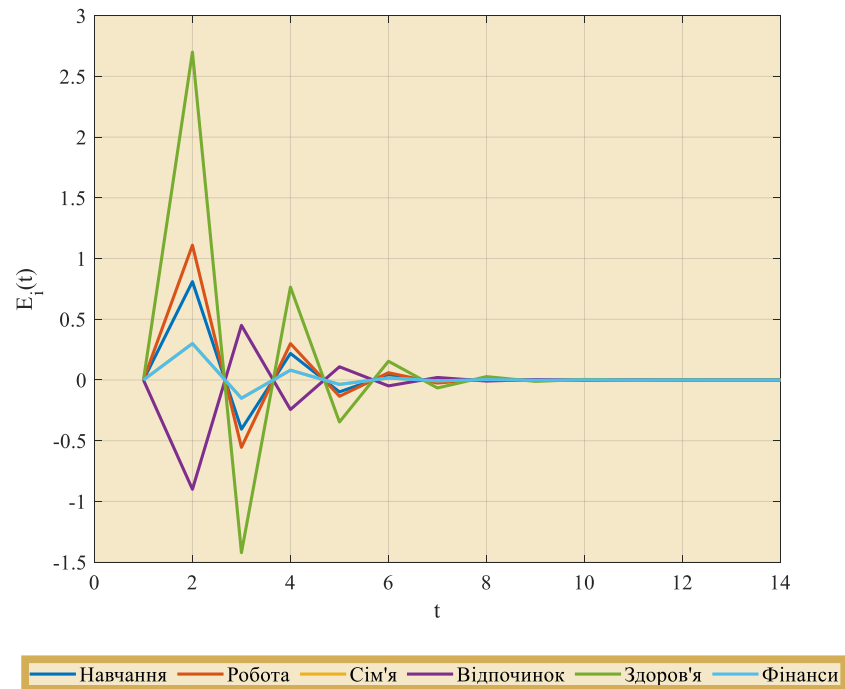


Рисунок 3.14

3.5 Висновки до розділу 3

На прикладі студента, що захворів на початку семестру, було продемонстровано, за допомогою моделі побудованої у розділі 2, динаміку його вільного руху соціально-навчального процесу. Результати моделювання показали, що якщо не впливати на систему, то студент наприкінці семестру матиме незадовільний стан у навчанні, що неодмінно призведе до негативних наслідків. Щоб запобігти такого розвитку подій, було введено регулятор стану який стабілізував нестійкий режим моделі, та на випадок невідомих параметрів моделі було застосовано алгоритми для їх оцінок. Хоча систему й було стабілізовано, але нові результати показали, що стан здоров'я студента не відновилось. Для вирішення цього питання було застосовано алгоритми

стабілізації вершин когнітивної карти на заданих рівнях, що було розглянуто у підрозділі 2.4. Нові результати показали задовільні результати.

РОЗДІЛ 4

ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

Функціонально-вартісний аналіз — це технологія, яка дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. Як прямі, так і побічні витрати розподіляються по продуктам та послугам у залежності від потрібних на кожному етапі виробництва обсягів ресурсів. Виконані на цих етапах дії у контексті метода ФВА називаються функціями.

Мета ФВА полягає у забезпеченні правильного розподілу ресурсів, виділених на виробництво продукції або надання послуг, на прямі та непрямі витрати. У даному випадку – аналізу функцій програмного продукту й виявлення усіх витрат на реалізацію цих функцій.

Такий аналіз дозволяє провести дослідження продукту заради його найоптимальнішої реалізації. Зокрема, виключення надлишкових або неефективних функцій дозволяє зменшити витрати на виробництво.

Фактично цей метод працює за таким алгоритмом:

а) визначається послідовність функцій, необхідних для виробництва продукту. Спочатку – всі можливі, потім вони розподіляються по двом групам: ті, що впливають на вартість продукту і ті, що не впливають. На цьому ж етапі оптимізується сама послідовність скороченням кроків, що не впливають на цінність і відповідно витрат;

б) для кожної функції визначаються повні річні витрати й кількість робочих часів;

в) для кожної функції на основі оцінок попереднього пункту визначається кількісна характеристика джерел витрат;

г) після того, як для кожної функції будуть визначені їх джерела витрат, проводиться кінцевий розрахунок витрат на виробництво продукту.

4.1 Постановка задачі проектування

У даному розділі ставиться задача оцінити можливості розробки програмного продукту для моделювання динаміки соціально-навчального процесу студента. Програма повинна реалізовувати зручний користувацький інтерфейс, можливості наглядної презентації даних, що обробились, корегування вхідних даних.

Відповідно цьому варто обирати і систему показників якості програмного продукту.

Технічні вимоги до продукту наступні:

- програмний продукт повинен функціонувати на персональних комп'ютерах із стандартним набором компонент;
- забезпечувати зручність і простоту взаємодії з користувачем;
- передбачати мінімальні витрати на впровадження програмного продукту;
- забезпечити високу швидкість обчислення результатів моделювання.

4.1.1 Обґрунтування функцій програмного продукту

Головна функція F_0 – розробка програмного продукту, який моделює динаміку соціально-навчального процесу студента. Виходячи з конкретної мети, можна виділити наступні основні функції ПП:

F_1 – вибір мови програмування;

F_2 – вибір середовища розробки;

F_3 – інформаційне вікно;

F_4 – інтерфейс користувача.

Кожна з основних функцій може мати декілька варіантів реалізації.

Функція F_1 :

а) мова програмування MATLAB;

б) мова програмування Python;

в) мова програмування C++.

Функція F_2 :

а) середовище MATLAB R2019a;

б) середовище Visual Studio 2019;

в) середовище Jupyter Notebook.

Функція F_3 :

а) виведення результатів роботи в окремий файл;

б) реалізація візуально-графічного модуля.

Функція F_4 :

а) інтерфейс користувача, наданий середовищем розробки;

б) розробка інтерфейсу за допомогою засобів середовища розробки.

4.1.2 Варіанти реалізації основних функцій

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 4.1). На основі цієї карти побудовано позитивно-негативну матрицю варіантів основних функцій (таблиця 4.1).

Морфологічна карта відображує всі можливі комбінації варіантів реалізації функцій, які складають повну множину варіантів ПП.

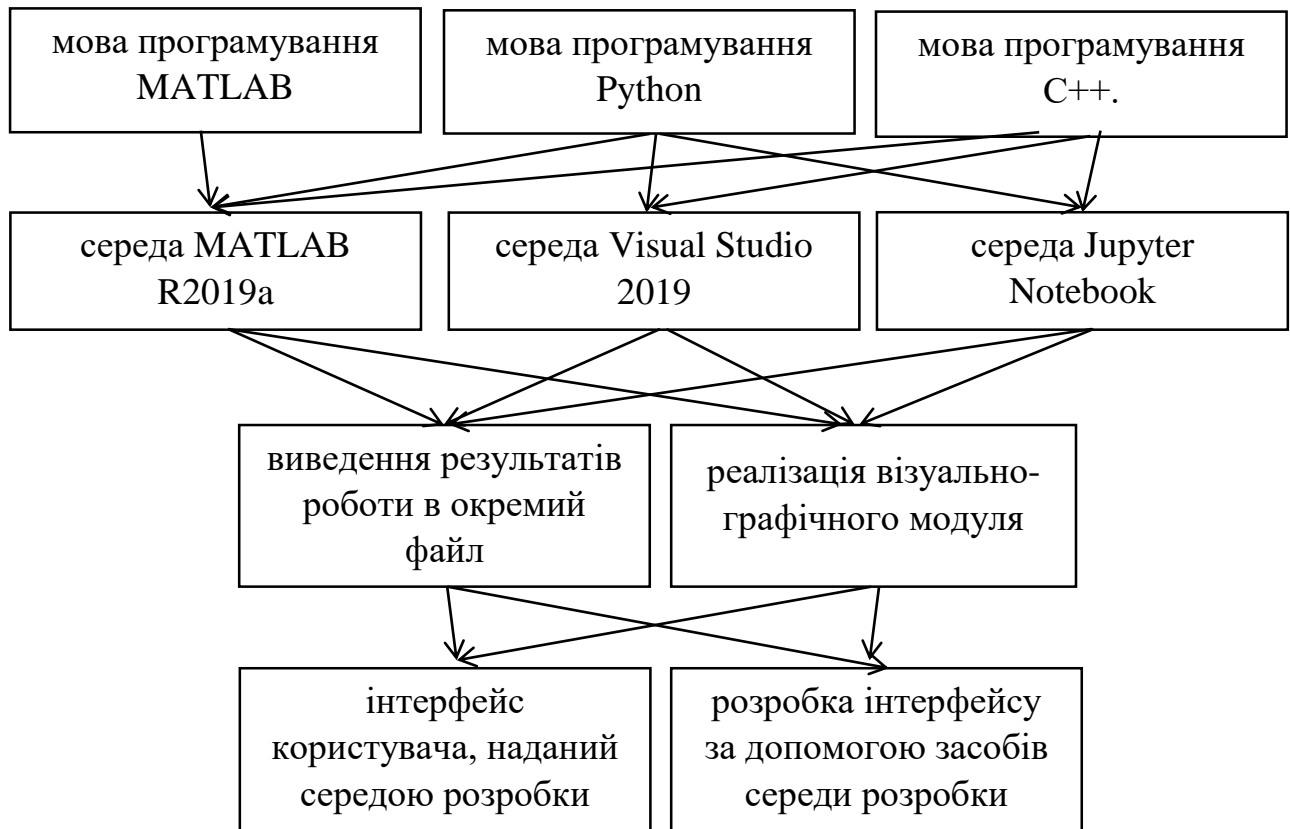


Рисунок 4.1

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому, що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція F1:

Оскільки розрахунки проводяться з великими об'ємами вхідних даних, то час написання програмного коду є дуже необхідним, тому варіант в) має бути відкинутий.

Таблиця 4.1

Основні функції	Варіанти реалізації	Переваги	Недоліки
F_1	А	Малий об'єм коду	Не кросплатформений
	Б	Кросплатформений	Швидкодія
	В	Швидкодія	Великий об'єм коду
F_2	А	Великий функціонал	Платна
	Б	Безкоштовна	Потребує багато пам'яті
	В	Проста у використанні	Малий функціонал
F_3	А	Збереження результатів	Важко аналізувати
	Б	Придатний вид для аналізу	Потребує додаткових засобів
F_4	А	Не потребує додаткових модулів	Більш складний для користування
	Б	Зручний у використанні	Необхідні додаткові модулі

Функція F2:

Оскільки написання ПП потребує багато різноманітного функціоналу середі розробки, то варіант в) можна відкинути.

Функція F3:

Оскільки вихідні дані повинні бути доступними та зрозумілими широкому колу користувачів, то варіант а) має бути відкинтий.

Функція F4:

Інтерфейс користувача не відіграє велику роль у даному програмному продукті, тому вважаємо варіанти а) та б) гідними розгляду.

Таким чином, будемо розглядати такі варіанти реалізації ПП:

1. $F1a - F2a - F3b - F4a$;
2. $F1a - F2a - F3b - F4b$.

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

4.2 Обґрунтування системи параметрів програмного продукту

4.2.1 Опис параметрів

На підставі даних про основні функції, що повинен реалізувати програмний продукт, вимог до нього, визначаються основні параметри виробу, що будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

- $X1$ – швидкодія мови програмування;
- $X2$ – об'єм пам'яті для збереження даних;
- $X3$ – час обробки даних;
- $X4$ – потенційний об'єм програмного коду.

$X1$: Відображає швидкодію операцій залежно від обраної мови програмування.

$X2$: Відображає об'єм пам'яті в оперативній пам'яті персонального комп'ютера, необхідний для збереження та обробки даних під час виконання програми.

$X3$: Відображає час, який витрачається на дії.

$X4$: Показує розмір програмного коду який необхідно створити безпосередньо розробнику.

4.2.2 Кількісна оцінка параметрів

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію ПП як показано у таблиці 4.2.

Таблиця 4.2

Назва Параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Швидкодія мови програмування	X1	Оп/мс	5000	12000	15000
Об'єм пам'яті для збереження даних	X2	Мб	64	32	16
Час обробки даних алгоритмом	X3	мс	1200	800	600
Потенційний об'єм програмного коду	X4	кількість строк коду	1200	750	300

За даними таблиці 4.2 будуються графічні характеристики параметрів – (рис. 4.2 – рис. 4.5).

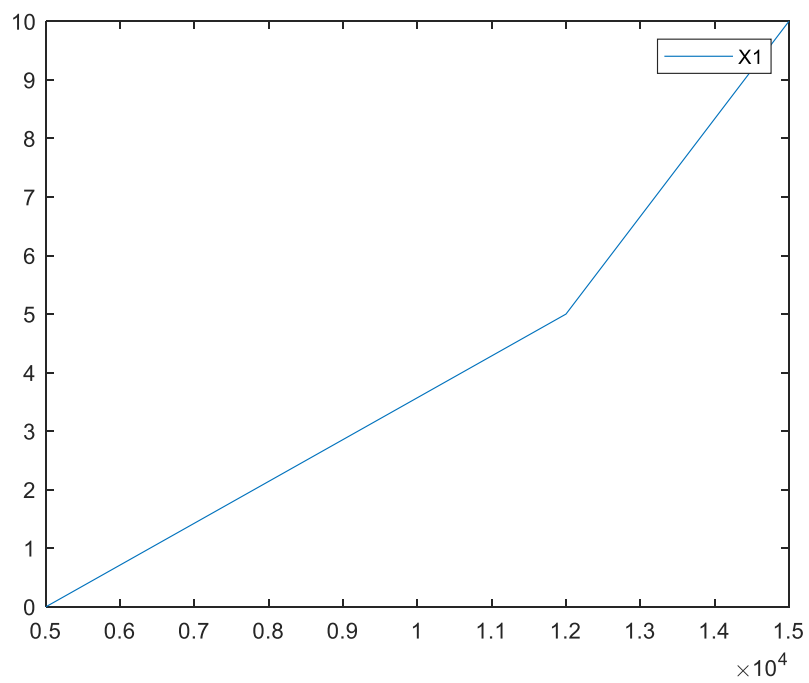


Рисунок 4.2

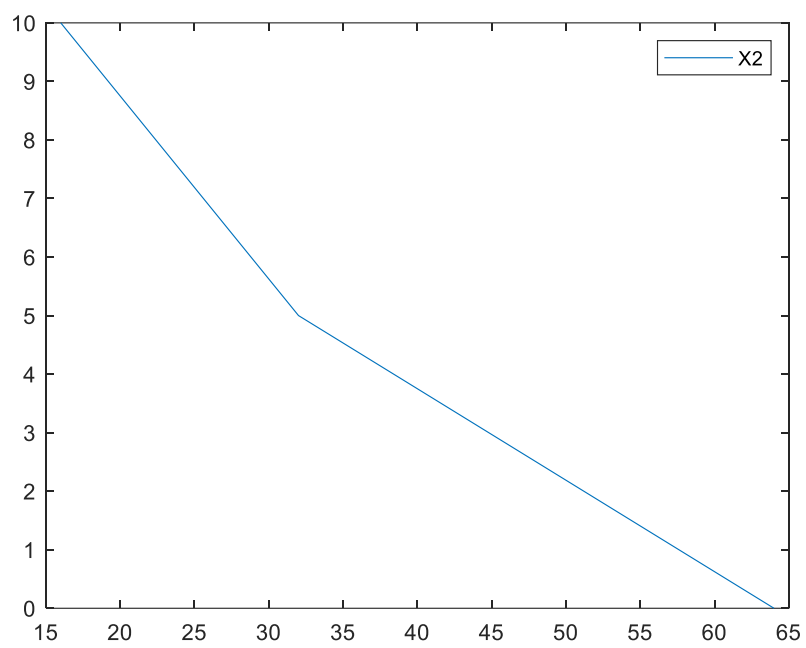


Рисунок 4.3

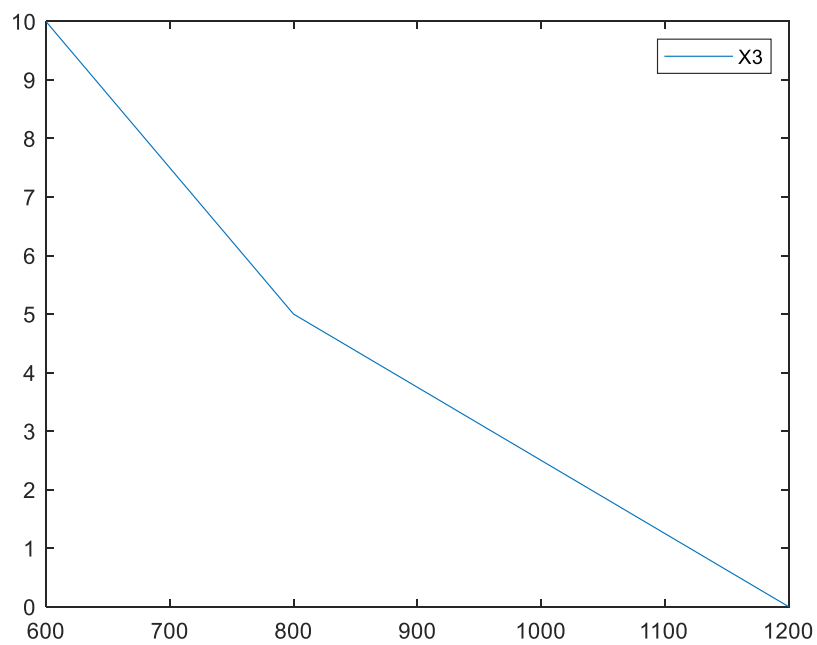


Рисунок 4.4

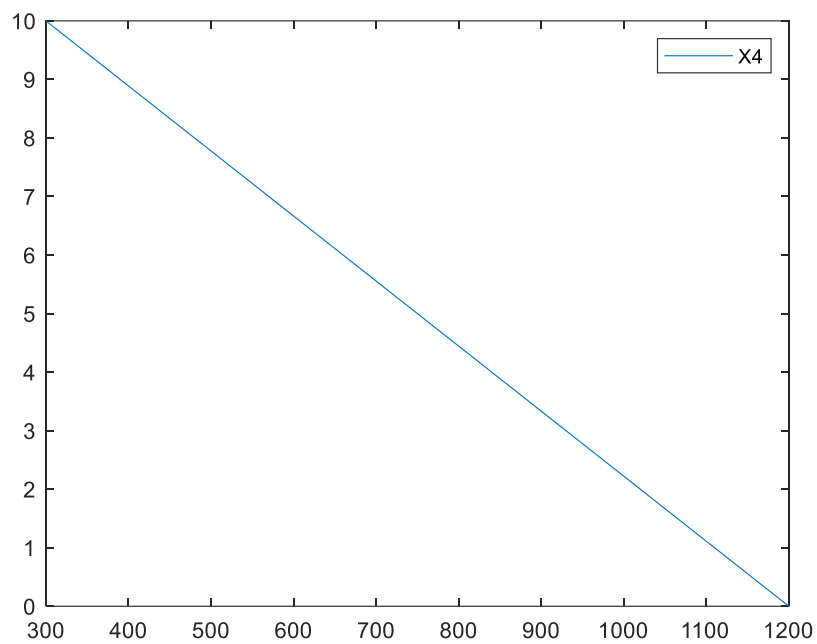


Рисунок 4.5

4.2.3 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту, який дає найбільш точні результати при знаходженні параметрів моделей адаптивного прогнозування і обчислення прогнозних значень.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 4.3.

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70, \quad (4.1)$$

де N – число експертів;

n – кількість параметрів;

R_{ij} – ранг.

Таблиця 4.3

Позначення параметра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів R_i	Відхилення Δ_i	Δ_i^2
			1	2	3	4	5	6	7			
$X1$	Швидкість мови програмування	Оп/мс	3	2	2	3	3	3	4	20	2,5	6,25
$X2$	Об'єм пам'яті для збереження даних	Мс	2	3	3	2	2	2	1	15	-2,5	6,25
$X3$	Потенційний об'єм програмного коду	кількість строк коду	1	1	1	1	1	1	2	8	-9,5	90,25
$X3$	Потенційний об'єм програмного коду	Кількість строк коду	4	4	4	4	4	4	3	27	9,5	90,25
	Разом		10	10	10	10	10	10	10	70	0	193

б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 17,5; \quad (4.2)$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

Числове значення, що визначає ступінь переваги i -го параметра над j -тим, a_{ij} визначається по формулі:

$$a_{ij} = \begin{cases} 1,5 & \text{при } X_i > X_j; \\ 1,0 & \text{при } X_i = X_j; \\ 0,5 & \text{при } X_i < X_j. \end{cases} \quad (4.6)$$

З отриманих числових оцінок переваги складемо матрицю $A = ||a_{ij}||$.

Для кожного параметра зробимо розрахунок вагомості K_{bi} за наступними формулами:

$$K_{bi} = \frac{b_i}{\sum_{i=1}^n b_i}, \quad (4.7)$$

де $b_i = \sum_{j=1}^N a_{ij}$;

a_{ij} - ступінь переваги i -го параметра над j -тим.

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{bi} = \frac{b'_i}{\sum_{i=1}^n b'_i}, \quad (4.8)$$

де $b'_i = \sum_{j=1}^N a_{ij} b_j$.

Як видно з таблиці 4.5, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 4.5

Параметри x_i	Параметри x_j				Перша ітер.		Друга ітер.	
	X1	X2	X3	X4	b_i	K_{Bi}	b_i^1	K_{Bi}^1
X1	1,0	1,5	1,5	0,5	4,5	0,281	20,25	0,281
X2	0,5	1,0	1,5	0,5	3,5	0,219	15,75	0,219
X3	0,5	0,5	1,0	0,5	2,5	0,156	11,25	0,156
X4	1,5	1,5	1,5	1,0	5,5	0,344	24,75	0,344
Всього:					16	1	72	1

4.3 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів X2 (об'єм пам'яті для збереження даних), X1 (швидкодія мови програмування), X3 (час обробки даних) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра X4 (потенційний об'єм програмного коду) обрано не найгіршим (не максимальним), тобто це значення відповідає або варіанту а) 650 строк коду або варіанту б) 1000 строк коду.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (табл. 4.6):

$$K_K(j) = \sum_{i=1}^n K_{ei,j} B_{i,j}, \quad (4.9)$$

де n – кількість параметрів;

K_{ei} – коефіцієнт вагомості i -го параметра;

B_i – оцінка i -го параметра в балах.

Таблиця 4.6

Основні функції	Варіант реалізації функції	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
$F1(X1)$	А	14000	9	0,281	2,529
$F2(X2)$	А	32	5	0,219	1,095
$F3(X3)$	Б	800	5	0,156	0,78
$F4(X4)$	А	650	9	0,344	3,096
	Б	1000	2	0,344	0,688

За даними з таблиці 4.6 за формулою

$$K_K = K_{\text{ТУ}}[F_{1k}] + K_{\text{ТУ}}[F_{2k}] + \dots + K_{\text{ТУ}}[F_{zk}], \quad (4.10)$$

де $K_{\text{ТУ}}[F_{ik}]$ – коефіцієнт рівня якості i функції.

Визначаємо рівень якості кожного з варіантів:

$$K_{K1} = 2,529 + 1,095 + 0,78 + 3,096 = 7,5,$$

$$K_{K2} = 2,529 + 1,095 + 0,78 + 0,688 = 5,092.$$

Як видно з розрахунків, кращим є перший варіант, для якого коефіцієнт технічного рівня має більше значення.

4.4 Економічний аналіз варіантів розробки ПП

Для визначення вартості використання програмного продукту спочатку проведемо розрахунок трудомісткості.

Всі варіанти включають в себе два окремих завдання:

1. Встановлення програмного продукту;
2. Використання програмного продукту;

Завдання 1 за ступенем новизни відноситься до групи А, завдання 2 – до групи Б. За складністю алгоритми, які використовуються в завданні 1 належать до групи 1; а в завданні 2 – до групи 3.

Для реалізації завдання 1 використовується довідкова інформація, а завдання 2 використовує інформацію у вигляді даних.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань.

Проведемо розрахунок норм часу на встановлення та використання для кожного з завдань. Загальна трудомісткість обчислюється як

$$T_0 = T_P \cdot K_{\Pi} \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М}, \quad (4.11)$$

де T_P – трудомісткість розробки програмного продукту;

K_{Π} – поправочний коефіцієнт;

$K_{СК}$ – коефіцієнт на складність вхідної інформації;

K_M – коефіцієнт рівня мови програмування;

$K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм;

$K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру ступеню новизни А та групи складності алгоритму 1, трудомісткість

дорівнює: $T_p = 90$ людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання: $K_{\Pi} = 1,5$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для всіх семи завдань рівний 1: $K_{СК} = 1$. Оскільки при розробці першого завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $K_{СТ} = 0,8$. Тоді, загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 90 \cdot 1,5 \cdot 0,8 = 108 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для подальших завдань.

Для другого завдання (використовується алгоритм третьої групи складності, степінь новизни Б), тобто $T_p = 30$ людино-днів, $K_{\Pi} = 0,7$, $K_{СК} = 1$, $K_{СТ} = 0,8$:

$$T_2 = 30 \cdot 0,7 \cdot 0,8 = 16,8 \text{ людино-днів.}$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$T_I = (108 + 16,8 + 4,8 + 16,8) \cdot 8 = 1171,2 \text{ людино-годин,}$$

$$T_{II} = (108 + 16,8 + 6,91 + 16,8) \cdot 8 = 1188,08 \text{ людино-годин.}$$

Найбільш високу трудомісткість має варіант II.

В розробці беруть участь два аналітика з окладом 9500 грн., один програміст з окладом 7000 грн. Визначимо зарплату за годину за формулою:

$$C_{\text{ч}} = \frac{M}{T_m \cdot t} \text{ грн.}, \quad (4.12)$$

де M – місячний оклад працівників;
 T_m – кількість робочих днів на тиждень;
 t – кількість робочих годин в день.

$$C_{\text{ч}} = \frac{9500 + 9500 + 7000}{3 \cdot 21 \cdot 8} = 51,5 \text{ грн.}$$

Тоді, розрахуємо заробітну плату за формулою

$$C_{\text{зп}} = C_{\text{ч}} \cdot T_i \cdot K_{\text{д}} \quad (4.13)$$

де $C_{\text{ч}}$ – величина погодинної оплати праці програміста;
 T_i – трудомісткість відповідного завдання;
 $K_{\text{д}}$ – норматив, який враховує додаткову заробітну плату.
 Зарплата розробників за варіантами становить:

$$\text{I.} \quad C_{\text{зп}} = 51,5 \cdot 1171,2 \cdot 1,2 = 72380,16 \text{ грн.};$$

$$\text{II.} \quad C_{\text{зп}} = 51,5 \cdot 1188,08 \cdot 1,2 = 73423,34 \text{ грн.}$$

Відрахування на єдиний соціальний внесок становить 22%:

$$\text{I.} \quad C_{\text{вд}} = C_{\text{зп}} \cdot 0,22 = 72380,16 \cdot 0,22 = 15923,63 \text{ грн.};$$

$$\text{II.} \quad C_{\text{вд}} = C_{\text{зп}} \cdot 0,22 = 73423,344 \cdot 0,22 = 16153,13 \text{ грн.}$$

Тепер визначимо витрати на оплату однієї машино-години ($C_{\text{м}}$).

Так як одна ЕОМ обслуговує одного аналітика з окладом 9500 грн., з коефіцієнтом зайнятості 0,41 то для однієї машини отримаємо:

$$C_{\Gamma} = 12 \cdot M \cdot K_3 = 12 \cdot 9500 \cdot 0,41 = 46740 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{3П} = C_{\Gamma} \cdot (1 + K_3) = 46740 \cdot (1 + 0,41) = 65903,4 \text{ грн.}$$

Відрахування на єдиний соціальний внесок:

$$C_{ВІД} = C_{3П} \cdot 0,22 = 65903,4 \cdot 0,22 = 14498,74 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 22000 грн.

$$C_A = K_{TM} \cdot K_A \cdot Ц_{ПР} = 1,15 \cdot 0,25 \cdot 22000 = 6325 \text{ грн.,}$$

де K_{TM} – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;

K_A – річна норма амортизації;

$Ц_{ПР}$ – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = K_{TM} \cdot Ц_{ПР} \cdot K_P = 1,15 \cdot 22000 \cdot 0,05 = 1265 \text{ грн.,}$$

де K_P – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$T_{\text{ЕФ}} = (D_{\text{К}} - D_{\text{В}} - D_{\text{С}} - D_{\text{Р}}) \cdot t_3 \cdot K_{\text{В}} \quad (4.14)$$

де $D_{\text{К}}$ – календарна кількість днів у році;

$D_{\text{В}}, D_{\text{С}}$ – відповідно кількість вихідних та святкових днів;

$D_{\text{Р}}$ – кількість днів планових ремонтів устаткування; t – кількість робочих годин в день;

$K_{\text{В}}$ – коефіцієнт використання приладу у часі протягом зміни.

$$T_{\text{ЕФ}} = (365 - 104 - 8 - 16) \cdot 8 \cdot 0,9 = 1706,4 \text{ годин},$$

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_{\text{С}} \cdot K_3 \cdot C_{\text{ЕН}}, \quad (4.15)$$

де $N_{\text{С}}$ – середньо-споживча потужність приладу;

K_3 – коефіцієнтом зайнятості приладу;

$C_{\text{ЕН}}$ – тариф за 1 КВт-годин електроенергії.

$$C_{\text{ЕЛ}} = 1706,4 \cdot 0,156 \cdot 0,61 \cdot 2,7515 = 446,791 \text{ грн.},$$

Накладні витрати розраховуємо за формулою:

$$C_{\text{Н}} = C_{\text{ПР}} \cdot 0,67 = 22000 \cdot 0,67 = 14740 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_{\text{А}} + C_{\text{Р}} + C_{\text{ЕЛ}} + C_{\text{Н}}. \quad (4.15)$$

$$C_{\text{ЕКС}} = 65903,4 + 14498,74 + 6325 + 1706,4 + 446,791 + 14740 = 103620 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 103620 / 1706,4 = 60,72 \text{ грн/час.}$$

Оскільки в даному випадку всі роботи, які пов'язані з використанням програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$C_{\text{М}} = C_{\text{М-Г}} \cdot T. \quad (4.16)$$

$$\text{I. } C_{\text{М}} = 60,72 \cdot 1171,2 = 71115 \text{ грн.};$$

$$\text{II. } C_{\text{М}} = 60,72 \cdot 1188,08 = 72140 \text{ грн.}$$

Накладні витрати складають 67% від заробітної плати:

$$C_{\text{Н}} = C_{\text{ЗП}} \cdot 0,67. \quad (4.17)$$

$$\text{I. } C_{\text{Н}} = 72380,16 \cdot 0,67 = 48494,7 \text{ грн.};$$

$$\text{II. } C_{\text{Н}} = 73423,34 \cdot 0,67 = 49193,6 \text{ грн.}$$

Отже, вартість розробки програмного продукту за варіантами становить:

$$C_{\text{ПП}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_{\text{М}} + C_{\text{Н}} \quad (4.18)$$

$$\text{I. } C_{\text{ПП}} = 72380,16 + 15923,63 + 71115 + 48494,7 = 207913,49 \text{ грн.};$$

$$\text{II. } C_{\text{III}} = 73423,34 + 16153,13 + 72140 + 49193,6 = 210910 \text{ грн.}$$

4.5 Вибір кращого варіанта програмного продукту техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{\text{TEP}j} = K_{\text{K}} / C_{\text{Ф}j}. \quad (4.19)$$

$$K_{\text{TEP}1} = 7,5 / 207913,49 = 36 \cdot 10^{-6};$$

$$K_{\text{TEP}2} = 5,092 / 210910 = 24,1 \cdot 10^{-6}.$$

Як бачимо, найбільш ефективним є перший варіант реалізації програми з коефіцієнтом техніко-економічного рівня $K_{\text{TEP}1} = 36 \cdot 10^{-6}$.

4.6 Висновки до розділу 4

В даній розрахунково-графічній роботі проведено повний функціонально-вартісний аналіз ПП, який було розроблено в рамках дипломного проекту. Процес аналізу можна умовно розділити на дві частини.

В першій з них проведено дослідження ПП з технічної точки зору: було визначено основні функції ПП та сформовано множину варіантів їх реалізації; на основі обчислених значень параметрів, а також експертних оцінок їх важливості

було обчислено коефіцієнт технічного рівня, який і дав змогу визначити оптимальну з технічної точки зору альтернативу реалізації функцій ПП.

Другу частину ФВА присвячено вибору із альтернативних варіантів реалізації найбільш економічно обґрунтованого. Порівняння запропонованих варіантів реалізації в рамках даної частини виконувалось за коефіцієнтом ефективності, для обчислення якого були обчислені такі допоміжні параметри, як трудомісткість, витрати на заробітну плату, накладні витрати.

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишились після першого відбору двох варіантів виконання програмного комплексу оптимальним є перший варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості $K_{\text{TEP}} = 36 \cdot 10^{-6}$.

Цей варіант реалізації програмного продукту має такі параметри:

- мова програмування – MATLAB;
- середовище розробки MATLAB R2019a;
- реалізація візуально-графічного модуля;
- інтерфейс, що наданий середовищем розробки.

Даний варіант виконання програмного комплексу дає користувачу зручний інтерфейс, потужний функціонал і швидкодію.

ВИСНОВКИ

Дана дипломна робота присвячена дослідженню соціально-навчального процесу студента за допомогою методів когнітивного моделювання та теорії керування.

В роботі було побудовано когнітивну карту соціально-навчального процесу студента, що представляє собою зважений орієнтований граф, вершини, якого, репрезентують різні фактори, з яких складається життя студента, а ребра – зв'язки між ними.

Дослідження імпульсного процесу КК показало, що модель є нестійкою. Для її стабілізації було розроблено систему керування з регулятором стану, що базується на еталонних моделях. На випадок невідомих параметрів моделі було розроблено систему стабілізації з адаптивним керуванням. Для стабілізації вершин КК на заданих рівнях було розроблено відповідну систему з регулятором, який, також, базується на еталонних моделях. Результати моделювання показали працездатність розроблених систем.

Для подальшого дослідження соціально-навчального процесу студента можна побудувати нову більш детальну когнітивну карту. Над ребрами надписувати не числа, а функції. Це дозволить більш точно відобразити взаємозв'язки між різними факторами у житті студента. Також можна реалізувати систему з модальним регулятором.

ЛИТЕРАТУРА

1. Axelrod R. The Structure of Decision: Cognitive Maps of Political Elites. Princeton University Press, 1976. 404 p.
2. Робертс Ф.С. Дискретные математические модели с приложениями к социальным, биологическим и экологическим задачам; пер. с англ. Москва: Наука, 1986. 496 с.
3. Авдеева З.К., Коврига С.В., Макаренко Д.И., Максимов В.И. Когнитивный подход в управлении / Проблемы управления. 2002. № 3. 2–8 с.
4. Максимов В.И. Структурно-целевой анализ развития социально-экономических ситуаций / Проблемы управления. 2005. № 3. 30 – 38 с.
5. Kosko B. Fuzzy Cognitive Maps / International Journal of Man-Machine Studies. 1986. № 24. 65–75 p.
6. Aguilar J. A Survey about Fuzzy Cognitive Maps Papers / International Journal of Computational Cognition. 2005. 3. № 2. 27–33 p.
7. Независимое экспертное обозрение. 2012. Вып. 4. 85 с. URL: <http://www.sovschola.ru/sites/default/files/userfiles/NEO/Vol.4.rar>.
8. Кулинич А.А. Когнитивные карты в поддержке принятия решений / Труды Конгресса по интеллектуальным системам и информационным технологиям «IS&IT'11». Москва: Физматлит, 2011. В 4 т., т. 1. 557 – 564 с.
9. Изерман Р. Цифровые системы управления; пер. с англ. М.: Мир, 1984. 541 с.
10. Григорьев В.В., Лукьянова Г.В., Сергеев К.А. Анализ систем автоматического управления: Учебное пособие. Санкт-Петербург.: СПбГУ ИТМО, 2009. 105 с.

11. Романенко В.Д., Милявский Ю.Л. Обеспечение устойчивости импульсных процессов в когнитивных картах на основе моделей в пространстве состояний / System Research & Information Technologies. 2014. № 1. 26 – 42 с.

12. Романенко В.Д., Милявский Ю.Л., Реутов А.А. Метод адаптивного управления / Проблемы управления и информатики. 2015. №2. 35-45 с.

ДОДАТОК А

ІЛЮСТРАТИВНИЙ МАТЕРІАЛ ДОПОВІДІ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ
ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

СИСТЕМИ СТАБІЛІЗАЦІЇ НЕСТІЙКИХ РЕЖИМІВ В СОЦІАЛЬНО-НАВЧАЛЬНОМУ ПРОЦЕСІ СТУДЕНТА

Виконав:

студент групи КА-54

Богдан Давид Романович

Керівник: проф., д.т.н.

Романенко Віктор Демидович



ОБ'ЄКТ, ПРЕДМЕТ І МЕТА ДОСЛІДЖЕННЯ

Об'єкт	<ul style="list-style-type: none"> • Соціально-навчальний процес студента
Предмет	<ul style="list-style-type: none"> • Нестійка когнітивна карта студента, та методи її стабілізації
Мета	<ul style="list-style-type: none"> • Розробка системи стабілізації нестійкого імпульсного соціально-навчального процесу студента



ПОСТАНОВКА ЗАДАЧІ

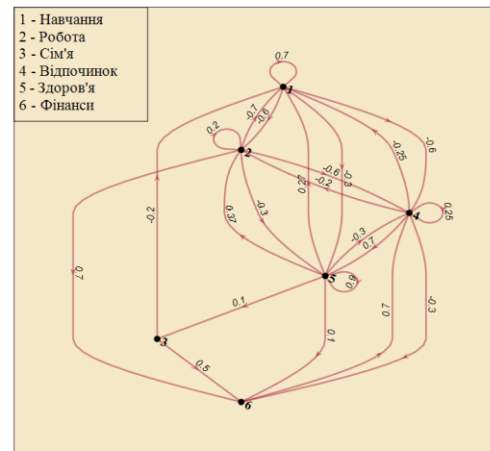
- Побудувати когнітивну карту соціально-навчального процесу студента;
- Розробити системи стабілізації:
 - Імпульсного процесу КК за допомогою регулятора стану, що базується на еталонних моделях:
 - При відомих параметрах моделі;
 - При невідомих параметрах моделі;
 - Вершин КК на заданих рівнях за допомогою регулятора стану, що базується на еталонних моделях ;
- Промодельовати:
 - Динаміку КК соціально-навчального процесу студента;
 - Стабілізацію імпульсного процесу КК за допомогою регулятора стану, що базується на еталонних моделях:
 - При відомих параметрах системи;
 - При невідомих параметрах системи;
 - Стабілізацію вершин КК на заданих рівнях за допомогою регулятора стану, що базується на еталонних моделях ;
- Зробити висновки.

3

КОГНІТИВНА КАРТА СОЦІАЛЬНО-НАВЧАЛЬНОГО ПРОЦЕСУ СТУДЕНТА

Покладемо такі вершини когнітивної карти соціально-навчального процесу студента:

- 1) навчання;
- 2) робота;
- 3) сім'я;
- 4) відпочинок;
- 5) здоров'я;
- 6) фінанси.



4

ІМПУЛЬСНИЙ ПРОЦЕС

$$\begin{pmatrix} \Delta Y_1(k+1) \\ \Delta Y_2(k+1) \\ \Delta Y_3(k+1) \\ \Delta Y_4(k+1) \\ \Delta Y_5(k+1) \\ \Delta Y_6(k+1) \end{pmatrix} = \begin{pmatrix} 0.7 & -0.6 & 0 & -0.6 & -0.3 & 0 \\ -0.7 & 0.2 & 0 & -0.6 & -0.3 & 0.7 \\ -0.2 & 0 & 0 & 0 & 0 & 0.5 \\ -0.25 & -0.2 & 0 & 2.5 & 0.7 & -0.3 \\ 0.27 & 0.37 & 0.1 & -0.3 & 0.9 & 0.1 \\ 0 & 0 & 0 & 0.7 & 0 & 0 \end{pmatrix}^T \begin{pmatrix} \Delta Y_1(k) \\ \Delta Y_2(k) \\ \Delta Y_3(k) \\ \Delta Y_4(k) \\ \Delta Y_5(k) \\ \Delta Y_6(k) \end{pmatrix}$$

Власні числа системи:

$$z_1 = -0.6753,$$

$$z_4 = 0.5309 - 0.7413i,$$

$$z_2 = 1.1702,$$

$$z_5 = 0.5174.$$

$$z_3 = 0.5309 + 0.7413i,$$

$$z_6 = -0.0241.$$

5

МОДЕЛЮВАННЯ ДИНАМІКИ КК СОЦІАЛЬНО-НАВЧАЛЬНОГО ПРОЦЕСУ СТУДЕНТА, ЩО ЗАХВОРІВ НА ПОЧАТКУ СЕМЕСТРУ

СИСТЕМА:

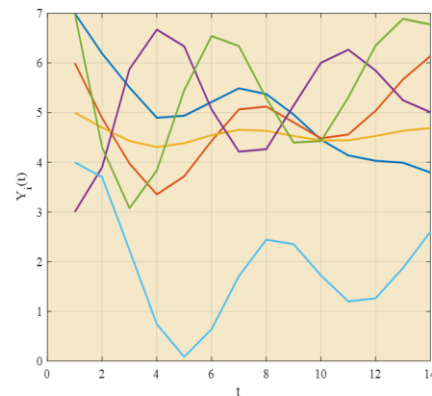
$$\Delta Y(k) = A \Delta Y(k-1).$$

ПОЧАТКОВІ ДАННІ:

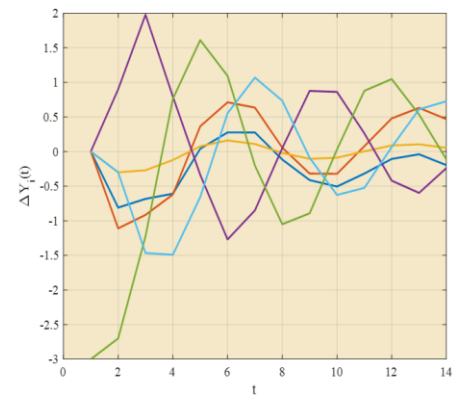
$$Y_0 = (7; 6; 5; 3; 7; 4),$$

$$\Delta Y_0 = (0; 0; 0; 0; -3; 0).$$

РЕЗУЛЬТАТ:



Навчання Робота Сім'я Відпочинок Здоров'я Фінанси



Навчання Робота Сім'я Відпочинок Здоров'я Фінанси

6

СИСТЕМА СТАБІЛІЗАЦІЇ ІМПУЛЬСНОГО ПРОЦЕСУ КК ЗА ДОПОМОГОЮ РЕГУЛЯТОРА СТАНУ, ЩО БАЗУЄТЬСЯ НА ЕТАЛОННИХ МОДЕЛЯХ

ПАРАМЕТРИ СИСТЕМИ ВІДОМІ

СИСТЕМА:

$$(I - A_1 z^{-1}) \Delta Y(k) = B_1 z^{-1} \Delta u(k)$$

ЗАКОН КЕРУВАННЯ:

$$\Delta u(k) = -D_0 (I + F_1 z^{-1})^{-1} \Delta Y(k)$$

ПАРАМЕТРИ ЗАКОНУ КЕРУВАННЯ:

$$F_1 = -A_1^{-1} A_{m2}, \quad D_0 = B_1^{-1} (A_1 + A_1^{-1} A_{m2} + A_{m1})$$

ПОЧАТКОВІ ДАНІ:

$$Y_0 = (7; 6; 5; 3; 7; 4), \Delta Y_0 = (0; 0; 0; 0; -3; 0),$$

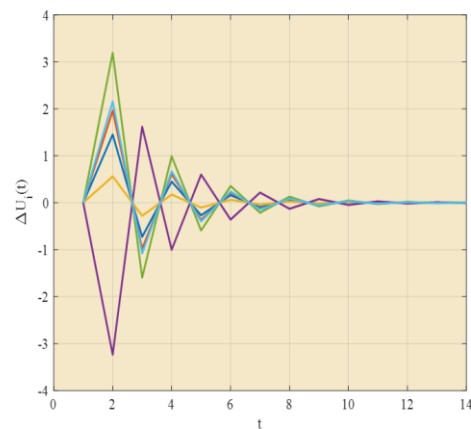
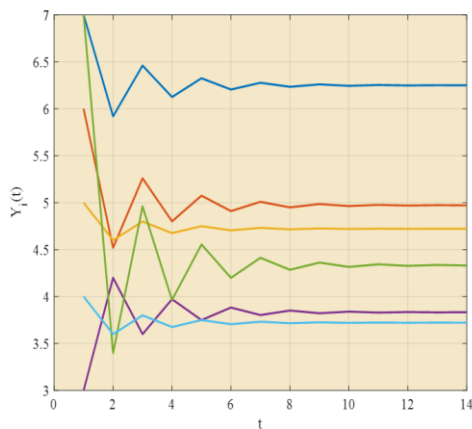
$$B_1 = I, z_1 = -0.6, z_2 = 0.1,$$

$$A_{m1} = 0.5I, \quad A_{m2} = -0.06I.$$

7

МОДЕЛЮВАННЯ СТАБІЛІЗАЦІЇ ІМПУЛЬСНОГО ПРОЦЕСУ КК ЗА ДОПОМОГОЮ РЕГУЛЯТОРУ СТАНУ, ЩО БАЗУЄТЬСЯ НА ЕТАЛОННИХ МОДЕЛЯХ

ПАРАМЕТРИ СИСТЕМИ ВІДОМІ



— Навчання — Робота — Сім'я — Відпочинок — Здоров'я — Фінанси

— Навчання — Робота — Сім'я — Відпочинок — Здоров'я — Фінанси

8

СИСТЕМА СТАБІЛІЗАЦІЇ ІМПУЛЬСНОГО ПРОЦЕСУ КК ЗА ДОПОМОГОЮ РЕГУЛЯТОРА СТАНУ, ЩО БАЗУЄТЬСЯ НА ЕТАЛОННИХ МОДЕЛЯХ

ПАРАМЕТРИ СИСТЕМИ НЕВІДОМІ

СИСТЕМА:

$$(I - A_1 z^{-1}) \Delta Y(k) = B_1 z^{-1} \Delta u(k)$$

ЗАКОН КЕРУВАННЯ:

$$\Delta u(k) = -D_0(I + F_1 z^{-1})^{-1} \Delta Y(k)$$

ПАРАМЕТРИ ЗАКОНУ КЕРУВАННЯ:

$$F_1 = F_1(k) = -\Theta(k)^{-1} A_{m2},$$

$$D_0 = D_0(k) = B_1^{-1}(\Theta(k) + \Theta(k)^{-1} A_{m2} + A_{m1})$$

ОЦІНКА ПАРАМЕТРІВ:

$$\Theta_i(k) = \Theta_i(k-1) + K_i(k) (\Delta Y_i(k) - b_i \Delta u_i(k-1) - X_i^T(k) \Theta_i(k-1))$$

$$K_i(k) = \frac{1}{1 + X_i^T(k) P_i(k-1) X_i(k)} P_i(k-1) X_i(k)$$

$$P_i(k) = P_i(k-1) - \frac{1}{1 + X_i^T(k) P_i(k-1) X_i(k)} P_i(k-1) X_i(k) X_i^T(k) P_i(k-1)$$

ПОЧАТКОВІ ДАНІ:

$$Y_0 = (7; 6; 5; 3; 7; 4),$$

$$\Delta Y_0 = (0; 0; 0; 0; -3; 0),$$

$$B_1 = I, z_1 = -0.6, z_2 = 0.1,$$

$$A_{m1} = 0.5I, A_{m2} = -0.06I,$$

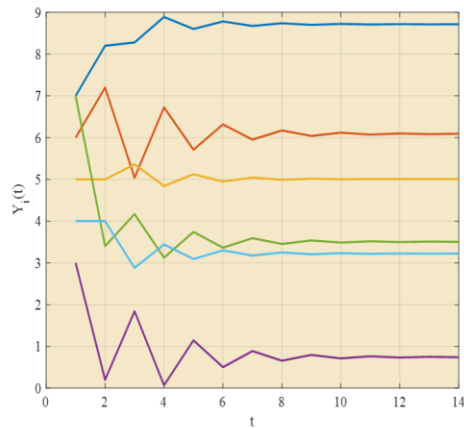
$$P_i(1) = 1000I,$$

$$\Theta_0 = \begin{pmatrix} 0.4 & -0.4 & 0 & -0.4 & -0.1 & 0 \\ -0.5 & 0.6 & 0 & -0.9 & -0.5 & 0.9 \\ -0.5 & 0 & 0 & 0 & 0 & 0.2 \\ -0.4 & -0.6 & 0 & 0.1 & 0.4 & -0.6 \\ 0.5 & 0.1 & 0.2 & -0.6 & 0.7 & 0.3 \\ 0 & 0 & 0 & 0.3 & 0 & 0 \end{pmatrix}^T$$

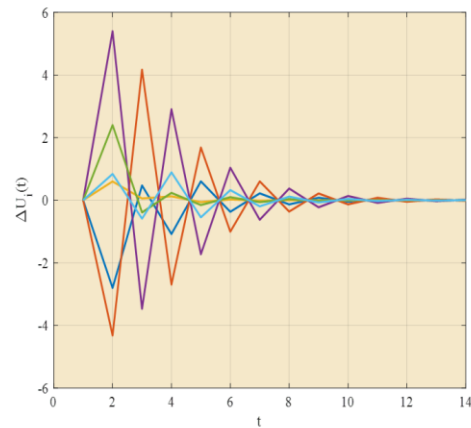
9

МОДЕЛЮВАННЯ СТАБІЛІЗАЦІЇ КК ЗА ДОПОМОГОЮ РЕГУЛЯТОРУ СТАНУ, ЩО БАЗУЄТЬСЯ НА ЕТАЛОННИХ МОДЕЛЯХ

ПАРАМЕТРИ СИСТЕМИ НЕВІДОМІ



— Навчання — Робота — Сім'я — Відпочинок — Здоров'я — Фінанси



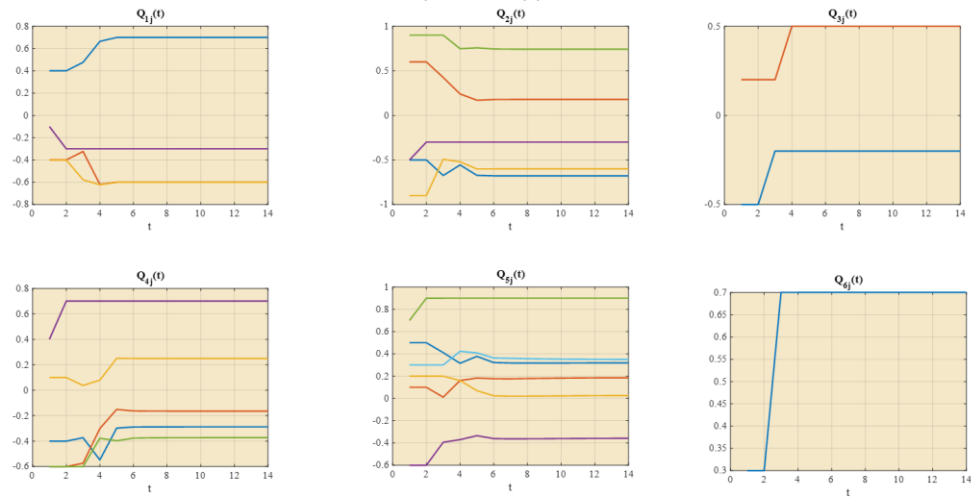
— Навчання — Робота — Сім'я — Відпочинок — Здоров'я — Фінанси

10

МОДЕЛЮВАННЯ СТАБІЛІЗАЦІЇ КК ЗА ДОПОМОГОЮ РЕГУЛЯТОРУ СТАНУ, ЩО БАЗУЄТЬСЯ НА ЕТАЛОННИХ МОДЕЛЯХ

ПАРАМЕТРИ СИСТЕМИ НЕВІДОМІ

ОЦІНКИ МОДЕЛІ



11

СИСТЕМА СТАБІЛІЗАЦІЇ ВЕРШИН КК НА ЗАДАНИХ РІВНЯХ ЗА ДОПОМОГОЮ РЕГУЛЯТОРА СТАНУ, ЩО БАЗУЄТЬСЯ НА ЕТАЛОННИХ МОДЕЛЯХ

СИСТЕМА:

$$(I - (I + A_1)z^{-1} + A_1z^{-2})Y(k) = B_1z^{-1}u(k)$$

ЗАКОН КЕРУВАННЯ:

$$u(k) = (D_0 + D_1z^{-1})(I + F_1z^{-1})^{-1}(G(k) - Y(k))$$

ПАРАМЕТРИ ЗАКОНУ КЕРУВАННЯ:

$$\begin{aligned} F_1 &= A_1^{-1}A_{m3} \\ D_0 &= B_1^{-1}(I + A_1 + A_{m1} - A_1^{-1}A_{m3}) \\ D_1 &= B_1^{-1}((I + A_1)A_1^{-1}A_{m3} - A_1 + A_{m2}) \end{aligned}$$

ПОЧАТКОВІ ДАННІ:

$$Y_0 = (7; 6; 5; 3; 7; 4),$$

$$\Delta Y_0 = (0; 0; 0; 0; -3; 0),$$

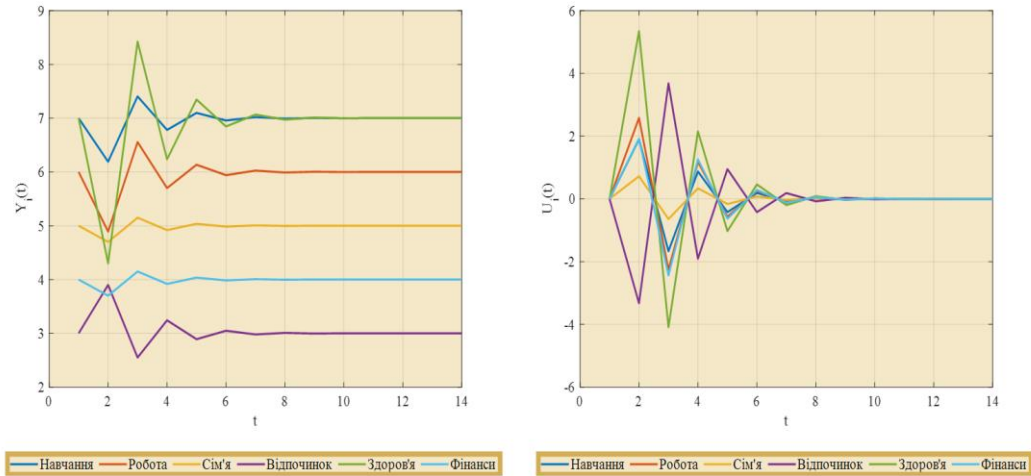
$$G = (7; 6; 5; 3; 7; 4),$$

$$B_1 = I, z_1 = -0.3, z_2 = 0.2, z_3 = -0.4$$

$$A_{m1} = 0.5I, \quad A_{m2} = -0.02I, \quad A_{m3} = -0.024I$$

12

МОДЕЛЮВАННЯ СТАБІЛІЗАЦІЇ ВЕРШИН КК НА ЗАДАНИХ РІВНЯХ ЗА ДОПОМОГОЮ РЕГУЛЯТОРА СТАНУ, ЩО БАЗУЄТЬСЯ НА ЕТАЛОННИХ МОДЕЛЯХ



13

ВИСНОВКИ

- Проведено дослідження підходів до когнітивного моделювання систем;
- Розглянуто методи переходу від когнітивної моделі до моделі у просторі станів і навпаки;
- Розроблено системи стабілізації нестійких імпульсних режимів КК, та стабілізації вершин КК на заданих рівнях;
- Промодельована динаміка КК студента, що захворів на початку семестру;
- Промодельована стабілізація КК студента, що захворів на початку семестру:
 - При відомих параметрах когнітивної карти;
 - При невідомих параметрах когнітивної карти;
- Промодельована стабілізація вершин КК студента на заданих рівнях;
- Проведено аналіз результатів.

14

ПЕРСПЕКТИВИ ЩОДО ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

- Використання функцій, в якості ваг ребер когнітивної карти, дозволить більш точно відобразити взаємозв'язки між різними вершинами когнітивної карти;
- Синтезувати модальний регулятор;
- Розробити більш детальну когнітивну модель.

15

ДЯКУЮ ЗА УВАГУ

16

ДОДАТОК Б

КОД ПРОГРАМНОГО ПРОДУКТУ

1. Моделювання вільного руху системи

```

A = sym([
    0.7000    -0.6000         0    -0.6000    -0.3000         0;
   -0.7000     0.2000         0    -0.6000    -0.3000     0.7000;
   -0.2000         0         0         0         0     0.5000;
   -0.2500    -0.2000         0     0.2500     0.7000    -0.3000;
    0.2700     0.3700     0.1000    -0.3000     0.9000     0.1000;
         0         0         0     0.7000         0         0]);

dY = [];
dY(:,1) = sym([0; 0; 0; 0; -3; 0]);

Y = [];
Y(:,1) = sym([7; 6; 5; 3; 7; 4]);

t = 14;

for i = 2:t
    dY(:,i) = A*dY(:,i-1);
    Y(:,i) = Y(:,i-1)+dY(:,i);
end

Y = double(Y)';
dY = double(dY)';

plY(Y)
plDY(dY)

```

2. Моделювання стабілізації системи з регулятором стану

```

A = [
    0.7000    -0.6000         0    -0.6000    -0.3000         0;
   -0.7000     0.2000         0    -0.6000    -0.3000     0.7000;
   -0.2000         0         0         0         0     0.5000;
   -0.2500    -0.2000         0     0.2500     0.7000    -0.3000;
    0.2700     0.3700     0.1000    -0.3000     0.9000     0.1000;
         0         0         0     0.7000         0         0];

```

```

Y1 = [];
Y1(1,:) = [7 6 5 3 7 4];

dY1 = [];
dY1(1,:) = [0 0 0 0 -3 0];

dU1 = [];
dU1(1,:) = [0 0 0 0 0 0];

U1 = [];
U1(1,:) = [0 0 0 0 0 0];

I = eye(6);

B = I;

z1 = -0.6;
z2 = 0.1;

Am1 = -(z1+z2)*I;
Am2 = z1*z2*I;

F = -A^(-1)*Am2;
D = B^(-1)*(A+A^(-1)*Am2+Am1);

t = 14;
for i = 2:t
    dY1(i,:) = A*dY1(i-1,:) + B*dU1(i-1,:);
    Y1(i,:) = Y1(i-1,:) + dY1(i,:);

    dU1(i,:) = -D*(F*D^(-1)*dU1(i-1,:) + dY1(i,:));
    U1(i,:) = U1(i-1,:) + dU1(i,:);
end

```

3. Моделювання стабілізації системи з адаптивним керуванням

```

A = [
    0.7000    -0.6000         0    -0.6000    -0.3000         0;
   -0.7000     0.2000         0    -0.6000    -0.3000     0.7000;
   -0.2000         0         0         0         0     0.5000;
   -0.2500   -0.2000         0     0.2500     0.7000   -0.3000;
    0.2700     0.3700     0.1000   -0.3000     0.9000     0.1000;
         0         0         0     0.7000         0         0];

Q0 = [
    0.4000   -0.4000         0   -0.4000   -0.1000         0;
   -0.5000     0.6000         0   -0.9000   -0.5000     0.9000;
   -0.5000         0         0         0         0     0.2000;

```

```

-0.4000    -0.6000         0     0.1000    0.4000    -0.6000;
 0.5000     0.1000     0.2000    -0.6000    0.7000     0.3000;
         0         0         0     0.3000         0         0];

I = eye(6);

B = I;

z1 = -0.6;
z2 = 0.1;

Am1 = -(z1+z2)*I;
Am2 = z1*z2*I;

t = 14;

U2 = [];
U2(1,:) = [0 0 0 0 0 0];

Y2 = [];
Y2(1,:) = [7 6 5 3 7 4];

dY2 = [];
dY2(1,:) = [0 0 0 0 -3 0];

dU2 = [];
dU2(1,:) = [0 0 0 0 0 0];

Q = [];
Q(:, :, 1) = Q0;

i1 = find(Q0(1, :));
i2 = find(Q0(2, :));
i3 = find(Q0(3, :));
i4 = find(Q0(4, :));
i5 = find(Q0(5, :));
i6 = find(Q0(6, :));

s1 = size(i1, 2);
s2 = size(i2, 2);
s3 = size(i3, 2);
s4 = size(i4, 2);
s5 = size(i5, 2);
s6 = size(i6, 2);

Q1 = [];
Q2 = [];
Q3 = [];
Q4 = [];
Q5 = [];
Q6 = [];
Q1(1, :) = Q(1, i1);

```

```

Q2(1,:) = Q(2,i2);
Q3(1,:) = Q(3,i3);
Q4(1,:) = Q(4,i4);
Q5(1,:) = Q(5,i5);
Q6(1,:) = Q(6,i6);

P1=[];
P2=[];
P3=[];
P4=[];
P5=[];
P6=[];
P1(:,:,1) = 1000*eye(size(i1, 2));
P2(:,:,1) = 1000*eye(size(i2, 2));
P3(:,:,1) = 1000*eye(size(i3, 2));
P4(:,:,1) = 1000*eye(size(i4, 2));
P5(:,:,1) = 1000*eye(size(i5, 2));
P6(:,:,1) = 1000*eye(size(i6, 2));

for i = 2:t
    dY2(i,:) = A*dY2(i-1,:) + B*dU2(i-1,:);
    Y2(i,:) = Y2(i-1,:) + dY2(i,:);

    X1(i,1:s1) = dY2(i-1, i1);
    X2(i,1:s2) = dY2(i-1, i2);
    X3(i,1:s3) = dY2(i-1, i3);
    X4(i,1:s4) = dY2(i-1, i4);
    X5(i,1:s5) = dY2(i-1, i5);
    X6(i,1:s6) = dY2(i-1, i6);

    P1(:,:,i) = P1(:,:,i-1)-(1/(1+X1(i,:)*P1(:,:,i-1)*X1(i,:)'))*P1(:,:,i-1)*X1(i,:)'*X1(i,:)*P1(:,:,i-1);
    P2(:,:,i) = P2(:,:,i-1)-(1/(1+X2(i,:)*P2(:,:,i-1)*X2(i,:)'))*P2(:,:,i-1)*X2(i,:)'*X2(i,:)*P2(:,:,i-1);
    P3(:,:,i) = P3(:,:,i-1)-(1/(1+X3(i,:)*P3(:,:,i-1)*X3(i,:)'))*P3(:,:,i-1)*X3(i,:)'*X3(i,:)*P3(:,:,i-1);
    P4(:,:,i) = P4(:,:,i-1)-(1/(1+X4(i,:)*P4(:,:,i-1)*X4(i,:)'))*P4(:,:,i-1)*X4(i,:)'*X4(i,:)*P4(:,:,i-1);
    P5(:,:,i) = P5(:,:,i-1)-(1/(1+X5(i,:)*P5(:,:,i-1)*X5(i,:)'))*P5(:,:,i-1)*X5(i,:)'*X5(i,:)*P5(:,:,i-1);
    P6(:,:,i) = P6(:,:,i-1)-(1/(1+X6(i,:)*P6(:,:,i-1)*X6(i,:)'))*P6(:,:,i-1)*X6(i,:)'*X6(i,:)*P6(:,:,i-1);

    K1(i,:) = (1/(1+X1(i,:)*P1(:,:,i-1)*X1(i,:)'))*P1(:,:,i-1)*X1(i,:);
    K2(i,:) = (1/(1+X2(i,:)*P2(:,:,i-1)*X2(i,:)'))*P2(:,:,i-1)*X2(i,:);
    K3(i,:) = (1/(1+X3(i,:)*P3(:,:,i-1)*X3(i,:)'))*P3(:,:,i-1)*X3(i,:);
    K4(i,:) = (1/(1+X4(i,:)*P4(:,:,i-1)*X4(i,:)'))*P4(:,:,i-1)*X4(i,:);
    K5(i,:) = (1/(1+X5(i,:)*P5(:,:,i-1)*X5(i,:)'))*P5(:,:,i-1)*X5(i,:);
    K6(i,:) = (1/(1+X6(i,:)*P6(:,:,i-1)*X6(i,:)'))*P6(:,:,i-1)*X6(i,:);

    Q1(i,:) = Q1(i-1,:)+K1(i,:)*(dY2(i,1)-B(1,1)*dU2(i-1,1)-X1(i,:)*Q1(i-1,:));

```

```

Q2(i,:) = Q2(i-1,:)+K2(i,:)*(dY2(i,2)-B(2,2)*dU2(i-1,2)-X2(i,:)*Q2(i-1,:))';
Q3(i,:) = Q3(i-1,:)+K3(i,:)*(dY2(i,3)-B(3,3)*dU2(i-1,3)-X3(i,:)*Q3(i-1,:))';
Q4(i,:) = Q4(i-1,:)+K4(i,:)*(dY2(i,4)-B(4,4)*dU2(i-1,4)-X4(i,:)*Q4(i-1,:))';
Q5(i,:) = Q5(i-1,:)+K5(i,:)*(dY2(i,5)-B(5,5)*dU2(i-1,5)-X5(i,:)*Q5(i-1,:))';
Q6(i,:) = Q6(i-1,:)+K6(i,:)*(dY2(i,6)-B(6,6)*dU2(i-1,6)-X6(i,:)*Q6(i-1,:))';

Q(1,i1,i) = Q1(i,:);
Q(2,i2,i) = Q2(i,:);
Q(3,i3,i) = Q3(i,:);
Q(4,i4,i) = Q4(i,:);
Q(5,i5,i) = Q5(i,:);
Q(6,i6,i) = Q6(i,:);

F = -Q(:, :, i)^(-1)*Am2;
D = B^(-1)*(Q(:, :, i)+Q(:, :, i)^(-1)*Am2+Am1);

dU2(i,:) = -D*(F*D^(-1)*dU2(i-1,:)' + dY2(i,:))';
U2(i,:) = U2(i-1,:) + dU2(i,:);
end

```

4. Моделювання стабілізації системи на заданих рівнях

```

A = sym([
    0.7000    -0.6000         0    -0.6000    -0.3000         0;
   -0.7000     0.2000         0    -0.6000    -0.3000     0.7000;
   -0.2000         0         0         0         0     0.5000;
   -0.2500   -0.2000         0     0.2500     0.7000   -0.3000;
    0.2700    0.3700    0.1000   -0.3000    0.9000    0.1000;
         0         0         0     0.7000         0         0]);

I = sym(eye(6));

B = I;

z1 = -0.3;
z2 = 0.2;
z3 = -0.4;

syms z

qwe = (z-z1)*(z-z2)*(z-z3);
dlqwe = diff(qwe);
d2qwe = diff(dlqwe);

Am1 = sym(subs(d2qwe,z,0)/2*I);
Am2 = sym(subs(dlqwe,z,0)*I);
Am3 = sym(subs(qwe,z,0)*I);

```

```

F1 = A^(-1)*Am3;

D0 = B^(-1)*(I+A+Am1-A^(-1)*Am3);
D1 = B^(-1)*((I+A)*A^(-1)*Am3-A+Am2);

syms z

D = det(I+F1*z);

O = (D0+D1*z)*(I+F1*z)^(-1)*D;

Y = [];
Y(:,1) = sym([7; 6; 5; 3; 7; 4]);
dY0 = sym([0;0;0;0;-3;0]);

G = sym([7; 6; 5; 3; 7; 4]);

E = [];
E(:,1) = G-Y(:,1);

U = [];
U(:,1) = sym([0; 0; 0; 0; 0; 0; 0]);

d1D = diff(D);
d2D = diff(d1D);
d3D = diff(d2D);
d4D = diff(d3D);
d5D = diff(d4D);
d6D = diff(d5D);

Au1 = -subs(d1D,z,0);
Au2 = -subs(d2D,z,0)/2;
Au3 = -subs(d3D,z,0)/2/3;
Au4 = -subs(d4D,z,0)/2/3/4;
Au5 = -subs(d5D,z,0)/2/3/4/5;
Au6 = -subs(d6D,z,0)/2/3/4/5/6;

d1O = diff(O);
d2O = diff(d1O);
d3O = diff(d2O);
d4O = diff(d3O);
d5O = diff(d4O);
d6O = diff(d5O);

Ae0 = subs(O,z,0);
Ae1 = subs(d1O,z,0);
Ae2 = subs(d2O,z,0)/2;
Ae3 = subs(d3O,z,0)/2/3;

```

```

Ae4 = subs(d40,z,0)/2/3/4;
Ae5 = subs(d50,z,0)/2/3/4/5;
Ae6 = subs(d60,z,0)/2/3/4/5/6;

t = 14;

i=2;

Y(:,i)=I*Y(:,i-1)+A*dY0+B*U(:,i-1);
E(:,i)=G-Y(:,i);
U(:,i)=Au1*U(:,i-1)+Ae0*E(:,i)+Ae1*E(:,i-1);

i=i+1;
Y(:,i) = (I+A)*Y(:,i-1)-A*Y(:,i-2)+B*U(:,i-1);
E(:,i) = G-Y(:,i);
U(:,i) = Au1*U(:,i-1)+Au2*U(:,i-2)+Ae0*E(:,i)+Ae1*E(:,i-1)+Ae2*E(:,i-2);

i=i+1;
Y(:,i) = (I+A)*Y(:,i-1)-A*Y(:,i-2)+B*U(:,i-1);
E(:,i) = G-Y(:,i);
U(:,i) = Au1*U(:,i-1)+Au2*U(:,i-2)+Au3*U(:,i-3)+Ae0*E(:,i)+Ae1*E(:,i-1)+Ae2*E(:,i-2)+Ae3*E(:,i-3);

i=i+1;
Y(:,i) = (I+A)*Y(:,i-1)-A*Y(:,i-2)+B*U(:,i-1);
E(:,i) = G-Y(:,i);
U(:,i) = Au1*U(:,i-1)+Au2*U(:,i-2)+Au3*U(:,i-3)+Au4*U(:,i-4)+Ae0*E(:,i)+Ae1*E(:,i-1)+Ae2*E(:,i-2)+Ae3*E(:,i-3)+Ae4*E(:,i-4);

i=i+1;
Y(:,i) = (I+A)*Y(:,i-1)-A*Y(:,i-2)+B*U(:,i-1);
E(:,i) = G-Y(:,i);
U(:,i) = Au1*U(:,i-1)+Au2*U(:,i-2)+Au3*U(:,i-3)+Au4*U(:,i-4)+Au5*U(:,i-5)+Ae0*E(:,i)+Ae1*E(:,i-1)+Ae2*E(:,i-2)+Ae3*E(:,i-3)+Ae4*E(:,i-4)+Ae5*E(:,i-5);

for i = 7:t
    Y(:,i) = (I+A)*Y(:,i-1)-A*Y(:,i-2)+B*U(:,i-1);
    E(:,i) = G-Y(:,i);

    U(:,i) = Au1*U(:,i-1)+Au2*U(:,i-2)+Au3*U(:,i-3)+Au4*U(:,i-4)+Au5*U(:,i-5)+Au6*U(:,i-6)+...
        Ae0*E(:,i)+Ae1*E(:,i-1)+Ae2*E(:,i-2)+Ae3*E(:,i-3)+Ae4*E(:,i-4)+Ae5*E(:,i-5)+Ae6*E(:,i-6);
end

Y = double(Y)';
E = double(E)';
U = double(U)';

```